



Université Blaise Pascal

MÉMOIRE

pour l'obtention d'une Habilitation à Diriger les Recherches

de L'Université Blaise Pascal
École Doctorale Sciences Pour L'Ingénieur

Discipline Informatique

présenté par

M. Marc Jaeger

Représenter et visualiser les objets naturels biologiques
De la plante au paysage

sous la responsabilité tutélaire du Professeur
David R.C. Hill

En soutenance publique le 15 décembre 2010 devant le jury composé de:

Monsieur Alain QUILLIOT, Professeur de l'Université Blaise Pascal, Clermont-Ferrand, Président
Monsieur David HILL, Professeur de l'Université Blaise Pascal, Clermont-Ferrand, Rapporteur
Madame Dominique Bechmann, Professeure de Université Louis Pasteur, Strasbourg, Rapportrice
Monsieur Fabrice NEYRET, Directeur de recherches du CNRS / INRIA Rhône Alpes, Rapporteur
Monsieur Jean-Pierre JESSEL, Professeur de l'Université Paul Sabatier, Toulouse, Examineur
Monsieur Philippe DE REFFYE, Directeur de Recherches du CIRAD, Montpellier, Examineur
Monsieur Baogang HU, Professeur de l'Académie des Sciences de Chine, Pékin, Invité



Université Blaise Pascal

MÉMOIRE

pour l'obtention d'une Habilitation à Diriger les Recherches

de L'Université Blaise Pascal
École Doctorale Sciences Pour L'Ingénieur

Discipline Informatique

présenté par

M. Marc Jaeger

Représenter et visualiser les objets naturels biologiques
De la plante au paysage

sous la responsabilité tutélaire du Professeur
David R.C. Hill

En soutenance publique le 15 décembre 2010 devant le jury composé de:

Monsieur Alain QUILLIOT, Professeur de l'Université Blaise Pascal, Clermont-Ferrand, Président
Monsieur David HILL, Professeur de l'Université Blaise Pascal, Clermont-Ferrand, Rapporteur
Madame Dominique Bechmann, Professeure de Université Louis Pasteur, Strasbourg, Rapportrice
Monsieur Fabrice NEYRET, Directeur de recherches du CNRS / INRIA Rhône Alpes, Rapporteur
Monsieur Jean-Pierre JESSEL, Professeur de l'Université Paul Sabatier, Toulouse, Examineur
Monsieur Philippe DE REFFYE, Directeur de Recherches du CIRAD, Montpellier, Examineur
Monsieur Baogang HU, Professeur de l'Académie des Sciences de Chine, Pékin, Invité

*A ma femme Randa,
et nos filles Laura, Elsa et Inès*

Remerciements

Mes premières pensées et mercis vont à Jean Françon, aujourd'hui en retraite bien méritée. Je lui dois la découverte du monde de la recherche, la rencontre avec les sciences du vivant, une certaine vision de la société, de la vie en général, avec un humanisme remarquable, et bien sûr, une direction de thèse riche en enseignements, discussions passionnées et passionnantes.

Je n'exprimerai jamais assez ma gratitude à Philippe de Reffye qui, dès les premiers jours de mon accueil à Montpellier en DEA n'a cessé d'alimenter et de me soutenir, tant dans l'environnement professionnel que dans ma vie personnelle. De l'encadrement de ma thèse il y a près de 25 ans à mon intégration au sein de l'EPI DigiPlante en 2007, jusqu'aujourd'hui, nos chemins se sont croisés, suivis, séparés, confondus, par périodes dans des projets qui nous ont conduit de Dallas à Tokyo, et dans une grande aventure en Chine, qui restera à jamais inscrite dans ma vie. Son recul, sa conduite de la recherche, sa créativité, son support, sa conduite des partenariats furent, sont et resteront à jamais pour moi même (et pour bien d'autres) des exemples et des modèles. Au delà du scientifique pionnier « le Maxwell des plantes » (sic M. Goursat de l'INRIA), je suis fier d'avoir pour ami et soutien un homme de grande valeur humaine et morale.

Merci à mes collègues, compagnons et amis du Cirad-Amap, et tout particulièrement à Philippe Borianne. Ensemble nous avons vécu une aventure scientifique passionnante sur les applications de l'imagerie volumique, nous conduisant du monde hospitalier, à celui des pépinières d'entreprises, en passant par celui de l'industrie agroalimentaire. Nous gardons tous deux de grands souvenirs avec des praticiens prestigieux, les Professeurs Henri Joyeux, François Bonnel, Guilhem Godlewski et Jacques Treil, que je tiens à absolument à citer ici tout en pensant à tous les autres, nombreux et qui, j'espère, ne m'en voudront pas de n'avoir été nommés.

Merci à mes collègues et amis d'Orient, du LIAMA, Zhang Xiaopeng, Baogang Hu, et de la CAU YanGuo et Zhang Baogui avec lesquels nous avons construit de bien belles choses et de grandes amitiés. C'est avec une fierté renouvelée que nous continuons à alimenter ce pont au quotidien, entre Pékin et Montpellier par nos échanges, nos accueils d'étudiants, nos retrouvailles régulières.

Merci aux étudiants que j'ai encadré et co-encadré jusqu'ici, Frédéric Banegas, Stéphane Chemouny, Zhi Feng Cai, Teng Jun, QingQiong Deng, Chu Zhao, Haiwen Wu, Vincent le Chevalier. Au delà des temps forts vécus pendant leurs thèses, ce sont de réelles complicités qui se sont créées, avec une fois de plus une grande fierté lorsque je considère aujourd'hui leurs parcours.

Tous mes remerciements à David Hill qui m'a beaucoup apporté et s'est beaucoup investi dans la finalisation de ce dossier.

Milles mercis aux rapporteurs et membres du jury qui ont bien voulu consacrer de leur précieux temps en ces périodes chargées.

Merci enfin à mes proches pour avoir supporté l'emprise passionnante de l'image et sa création qui m'ont inmanquablement conduit à des maladresses, oublis.... Je resterai à jamais surpris de leur patience et amour.

Marc Jaeger, le 8 septembre 2010

Sommaire

Remerciements.....	7
Sommaire.....	9
Rapport Scientifique.....	13
Introduction.....	14
Le contexte: La représentation numérique visuelle d'objets naturels	15
Représentation et visualisation.....	15
Les finalités de la visualisation des plantes et cultures.....	16
Les domaines d'application concernés par la visualisation des plantes, cultures et paysages. .	16
De l'image statique vers la réalité virtuelle et la représentation du fonctionnement.....	17
De la représentation explicite de l'organe à celle du fonctionnement du paysage.....	18
Les axes de recherches à développer pour la visualisation des plantes et paysages.....	19
CHAPITRE I. La représentation de la structure des végétaux et de structures anatomiques.....	21
La modélisation et la simulation de la géométrie des plantes simulées.....	22
Plantes virtuelles: quelques approches.....	22
Les modèles Amap et GreenLab.....	23
La simulation de la structure topologique.....	23
La simulation de la structure géométrique.....	23
La représentation de la géométrie dans les modèles GreenLab.....	24
La visualisation de plantes et de paysages et leurs développements techniques.....	25
Les reconstructions géométriques de plantes et structures anatomiques.....	29
Des structures ramifiées végétales à l'imagerie médicale ?	29
De la visualisation à la caractérisation avec l'imagerie volumique.....	29
La décomposition hiérarchique d'objets ramifiés dans un espace voxel	32
Les apports de l'imagerie volumique pour la représentation des plantes.....	33
Les modèles de plantes définis par reconstruction 3D.	34
La construction des couronnes végétales à partir de points laser.....	34
Les développements en cours autour de la reconstruction 3D laser.....	35
Plantes simulées ou reconstruites: une même représentation géométrique.....	37
CHAPITRE II. Vers la visualisation réaliste temps réel des plantes et paysages.....	38
La représentation multi-échelle polygonales des plantes simulées	39
Modèles et visualisation multi-échelle des systèmes ramifiés (branches).....	39
La représentation et la visualisation multi-échelle des organes (feuilles).....	42
Vers la représentation multi-échelle temps réel de forêts.....	45
Les limites des approches polygonales multi-échelles	47

Les représentations multi-échelles basées image.....	47
Imposteurs.....	47
Représentations issues de plongements dans l'espace voxel.....	49
 L'évolution des approches LoD pour la visualisation des plantes et des paysages.....	51
<u>CHAPITRE III.</u> Les paysages dits fonctionnels.....	52
De la plante à la culture et au paysage avec un même modèle de plante ?.....	53
Un système complexe.....	53
Les approches originales de la communauté graphique.....	54
La définition de "biotopes".....	54
Représenter et simuler la compétition.....	54
Simulation et visualisation de la compétition avec GreenLab.....	55
 Le développement d'outils de simulation et visualisation de paysages fonctionnels.....	56
Une première approche dans un espace voxel.....	56
Un nouveau prototype pour la simulation de paysages.....	58
La formalisation autour du paysage fonctionnel.....	60
La synchronisation des modèles	60
La modélisation de la compétition par les ressources.....	61
Les développements à venir.....	62
 <u>CHAPITRE IV.</u> Vers de nouvelles méthodologies pour représenter et visualiser les plantes.....	63
L'emploi des techniques GPU.	65
La génération de la géométrie des plantes à partir de leur squelette.....	65
De l'usage des sous-structures.....	66
 Les limites des approches multi-échelles et de l'instanciation.....	67
Une nouvelle approche pour créer des représentations géométriques.....	67
La reconstruction de la géométrie à partir du graphe de simulation	67
La reconstruction de géométries d'étapes de croissances passées	68
Reconstruire une culture, une forêt, un paysage à partir d'un unique individu ?.....	69
 Vers une représentation géométrique de la production des plantes.....	71
 L'intégration WEB. Le futur des applications de recherche sur l'environnement ?	72
 <u>CONCLUSION</u>	75
 <u>Index des figures</u>	80
 <u>Références</u>	82
 <u>Annexe 1.</u> Sélection de publications.....	93

**Représenter et visualiser les objets naturels biologiques.
De la plante au paysage.**

Mémoire pour la soutenance d'une Habilitation à Diriger des Recherches

Marc Jaeger. Septembre 2010

Introduction

Le contexte: La représentation numérique visuelle d'objets naturels

Le développement des techniques de l'informatique et des communications ont bouleversé nos sociétés. Mais la révolution de la représentation numérique reste à venir. Demain, sur nos mobiles, dans nos maisons, dans les lieux publics, nous aurons des représentations virtuelles tridimensionnelles interactives mêlant images réelles, éléments simulés sur lesquels on pourra interagir. Dès aujourd'hui, dans le monde industriel, l'usage de maquettes construites/partagées par divers corps de métiers se répand; au quotidien nos enfants jouent sur des jeux dont les réalismes visuels et comportementaux deviennent inquiétants; et la télévision 3D – connectable sur le Web, fait son apparition dans nos salons.

Mais, pour l'heure, la représentation visuelle des objets naturels et vivants demeure un défi. Il est indéniable que de nombreux progrès ont été accomplis depuis l'apparition de la synthèse d'image. On peut aujourd'hui considérer que l'on peut représenter n'importe quel objet naturel en 3D avec un grand réalisme. Le faire rapidement, représenter ses évolutions, ses dynamiques, interagir avec lui est une toute autre chose. A la représentation de la structure -son architecture pour prendre le terme consacré aux plantes-, il faut inclure des accès à des fonctions qui pilotent sa dynamique.

En cela, travailler sur la représentation des végétaux semble à priori, plus aisé. Car l'architecture d'une plante est la résultante de la dynamique de croissance en interaction avec l'environnement. En fait, il n'en est rien; car d'une part, il n'est pas indispensable de modéliser cette dynamique pour obtenir des représentations, et d'autre part, cette définition, ouverte, peut mener à une complexité totalement rédhibitoire.

Représentation et visualisation

L'obtention d'une image d'un végétal ou d'un paysage simulé sur un support tel qu'un écran obéit au même schéma que l'obtention d'une image d'un objet dit artificiel. La création des images par les techniques d'infographie 3D s'inscrit classiquement dans un enchaînement d'étapes :

- la modélisation des objets de la scène en trois dimensions,
- la définition des paramètres de vue,
- la définition des paramètres et modèles d'éclairage,
- les créations et les affectations des matériaux de surface et des textures,
- la définition du mode de rendu et le calcul des images proprement dit.

Dans notre cas, les étapes de modélisations et de rendu sont celles qui présentent des spécificités. La modélisation car la définition de la géométrie des objets est particulière. Le rendu car le volume en cause est important.

La modélisation couvre en fait de multiples aspects avant d'aboutir à une description géométrique tridimensionnelle. Sous le terme de **représentation** (d'une plante) nous désignons une approche permettant de calculer une description géométrique tridimensionnelle de l'objet d'intérêt, dans le but de le visualiser. Cette représentation peut être proche d'une modélisation géométrique (par exemple un simple polygone destiné à porter une texture image d'un arbre). Nous verrons qu'en général, nous travaillerons avec des approches de plus haut niveau, par exemple en limitant la description aux positions et orientations des composants constituant le végétal, voir à un simple graphe valué qui requiert un plongement géométrique pour obtenir une instance tridimensionnelle de l'objet.

Réciproquement, nous désignons sous le terme **visualisation** les techniques intervenant plus particulièrement dans les étapes de rendu. La visualisation sous-entend que le plongement géométrique des constituants des scènes, les conditions d'éclairage et de vue soient définis.

Ici, l'effort est bien sûr concentré sur la représentation, les développements de visualisation relèvent plutôt de l'ingénierie, mais ne peuvent être négligés pour autant.

Les finalités de la visualisation des plantes et cultures

A l'UMR Amap du Cirad, nous cherchons en fait simplement à contribuer à la mise en place de modèles utiles et opérationnels en agronomie, foresterie. Pour et avec nos collègues modélisateurs, nos développements sur la visualisation des plantes et cultures visent principalement à :

- montrer des structures calculées de plantes
- visualiser des résultats de processus d'établissement des structures
- visualiser rapidement des plantes seules et des cultures, à faible coût, sur du matériel usuel
- pouvoir interagir interactivement avec les structures simulées
- représenter des fonctionnements
- étudier qualitativement les relations structure/fonction
- interagir sur les paramètres du fonctionnement et observer les changements
- ...

En fait, le besoin est bien plus large, couvrant un large panel d'applications.

Les domaines d'application concernés par la visualisation des plantes, cultures et paysages

Avec un recul de plus de 20 ans depuis l'apparition des modèles de simulation de la structure des plantes, la représentation et la visualisation 3D des plantes, cultures et paysages ont montré leurs intérêts, couvrant des aspects variés et souvent connexes. Dans nos travaux à l'UMR Amap, on peut, sans être exhaustif, citer quelques domaines d'applications:

- La validation qualitative des simulations fut une préoccupation de première heure [MP_deReffye988b], [MP_Homazi000] et reste importante aujourd'hui [MSL_Cournede010]. La structure simulée semble-t-elle juste ? correcte ? Elle fonde la crédibilité même du modèle aux yeux d'un vaste public, spécialiste ou non.
- La communication entre spécialistes est facilitée, en particulier grâce à l'utilisation de maquettes tridimensionnelles, par exemple pour des simulations d'interceptions lumineuses [MP_Dauza008] ou de propagations de feux.
- L'enseignement auprès des jeunes et moins jeunes en permettant de visualiser les comportements d'un modèle dynamique et d'aider à comprendre les principes sous-jacents [ST_Jaeger991] [MP_deReffye008b]
- La reconstruction de flores, voire de paysages passés [MP_Davie000] (Figure Int_1)
- La représentation permet d'extraire l'individu de son contexte, et si elle est suffisamment réaliste pour identifier une espèce, voire un clone, elle peut être un élément d'apprentissage. Citons par exemple, la modélisation de collections de Bégonias [MP_Lecoustre992] ou la

modélisation et simulation de clones d'hévéa représentés sur des plaquettes couleurs destinées à aider les producteurs d'hévéa dans leur reconnaissance des clones en milieu réel. [MP_Dinouard987], [MP_Dinouard988]

- La communication avec le public autour des applications de projets d'aménagements couvrant un aspect paysager dans le cas de réalisations architecturales ou plus généralement d'infrastructures [MSL_Edelin989], [MP_deReffye988c], [ST_Bionatics010], [CG_Jaeger010]



Figure Int_1 Reconstruction d'un paysage du carbonifère (auteur: R. Lecoustre [MP_Davie000])

- Et bien sur les représentations pour le spectacle et les applications multimédia pour lesquels l'implication des scènes végétales constituent un élément de réalisme contextuel psychologique important [MP_deReffye988a], [MP_Blaise998], [MP_Sun009], [MP_Jaeger010].

De l'image statique vers la réalité virtuelle et la représentation du fonctionnement

La simulation de la géométrie des plantes est lourde et complexe car devant assurer simultanément une représentation hiérarchique ramifiée, structurée, organisée (celle du "bois") et une représentation d'une géométrie tout aussi complexe mais de nature totalement distincte, éparse, distribuée, s'inscrivant dans une dynamique temporelle courte, aux aspects visuels riches en couleurs, formes et dynamiques (celle des organes: feuilles, fleurs, fruits). Ces deux représentations soulèvent donc de nombreuses questions de recherche dans un contexte de visualisation réaliste et rapide (temps réel).

La définition de modèle de représentation de végétaux multi-échelle pour la représentation réaliste et/ou temps réel de plantes, cultures, forêts, paysages reste aujourd'hui un domaine de recherche très actif [MP_Godin000]. A cet aspect de réalisme visuel s'ajoute un nouveau besoin: celui d'un réalisme de situation voire comportemental. Aujourd'hui, la plupart des applications multimédia grand public ou industrielles impliquent des interfaces utilisateurs plus ou moins complexes permettant d'interagir avec les utilisateurs. La réalité virtuelle et, avec le développement des technologies d'acquisition d'images et de reconstruction, la réalité augmentée, ne se cantonnent plus à des applications ludiques, mais deviennent des outils d'échange, de communication, voire de design dans le monde industriel. Les représentations des objets "manipulables" s'adossent sur des modèles de description impliquant des simulations dynamiques (souvent mécaniques).

Grâce au développement des technologies des cartes graphiques (GPU) et des approches de rendu orientée images, de multiples contributions permettent aujourd'hui la représentation de scènes naturelles plus ou moins homogènes en temps réel. Dans la plupart des applications, et de manière systématique dans le multimédia, la recherche de la performance graphique est avant tout fondée sur des critères de fluidité du parcours dans les scènes. La quête du réalisme visuel ne s'appuie pas sur la représentation réaliste des composants de sa scène mais sur l'aptitude à plonger le spectateur dans un contexte, une atmosphère. Avec des modèles d'illumination ad-hoc, des traitements d'ombrage approximatif, l'usage d'imposteurs est convaincant. Ce réalisme orienté parcours visuel ne privilégie pas la dynamique intrinsèques aux systèmes, tant à l'échelle de la plante qu'à l'échelle de la scène.

Dans notre contexte, celui de la recherche agronomique, avec la pression sociétale croissante autour du changement climatique, du développement durable et d'une relative prise de conscience de l'importance de l'environnement avec ses valeurs écologiques, ces besoins de représentation de la dynamique sont de fait extrêmement complexes et lourds. Il faut non seulement obtenir des représentations qui présentent une cohérence "écologique" mais disposant aussi de capacités d'évolutions fondées sur des transformations plausibles.

Comment peut-on développer des approches autorisant le maintien des performances acquises aujourd'hui tout en permettant d'introduire des dynamiques complexes et réalistes dans les scènes ?

Le pari sur l'évolution des technologies ne sera pas suffisant pour affronter l'évolution de la complexité. L'idée phare que nous poussons ici est de construire simplement sur la base d'un compromis entre réalisme comportemental et réalisme visuel (statique). En fait, dans une application donnée, ce n'est pas d'un compromis dont il s'agit mais plutôt de la recherche d'un équilibre: entre une description à une échelle dite fine (vue de près) à une description à une échelle globale (vue satellitaire). De l'échelle dite fine à l'échelle dite, globale, cette description présente une part décroissante de représentation de la structure pour une part croissante de représentation du fonctionnement.

De la représentation explicite de l'organe à celle du fonctionnement du paysage

En effet, à l'échelle locale de l'organe, la représentation visuelle ne peut qu'être explicite, donc géométrique, précise tant sur la structure que son aspect (texture). Nous sommes ici dans un contexte "classique" de synthèse d'image centré sur les propriétés -de surface- de l'objet. A l'opposé, par exemple sur une vue aérienne, voire satellite, la représentation visuelle est principalement fonction de la présence de biomasse (pour des caractéristiques de géographie physique homogènes). Nous sommes cette fois dans un contexte "classique" de traitement du signal échantillonné sur une grille bidimensionnelle. La représentation est non pas attachée à l'objet végétal mais au terrain (l'espace) qui présente des propriétés radiatives relatives au fonctionnement de la végétation.

De même, dans nos besoins de visualisations, les aspects fonctionnels influent peu sur la représentation de l'organe (en écartant la taille et l'aspect de surface), et réciproquement, il est totalement inutile de s'intéresser à la représentation de la structure dans l'image aérienne.

A ce gradient structurel/fonctionnel nous pouvons ajouter un gradient temporel. En effet, lorsque la

visualisation est centrée sur la plante ou ses organes, on est facilement amené à se déplacer pour visualiser le voisinage et effectuer des déplacements dans l'espace autour de l'objet d'intérêt. La plante est dans ce cas statique (du moins en terme d'évolution), il n'y a pas besoin de solliciter de nouvelles simulations et plongements géométriques. A l'inverse, à l'échelle du paysage, on est plus facilement amené à considérer des dynamiques temporelles sur un même site. On centre aussi son attention sur des zones plus précises. Dans les deux cas, il faut effectuer des calculs importants: sur la dynamique de production ou sur la génération de structures géométriques.

Les axes de recherches à développer pour la visualisation des plantes et paysages

En affinant ces gradients, on peut identifier des niveaux de représentations et des techniques, qu'il faudrait idéalement maîtriser pour disposer de représentations réalistes et performantes des plantes, cultures et paysages virtuels. La définition de ces niveaux conduit à décrire des thématiques de recherche et développement autour de la simulation et de la visualisation des plantes et paysages, ce que nous avons fait au LIAMA (Figure Int_2)

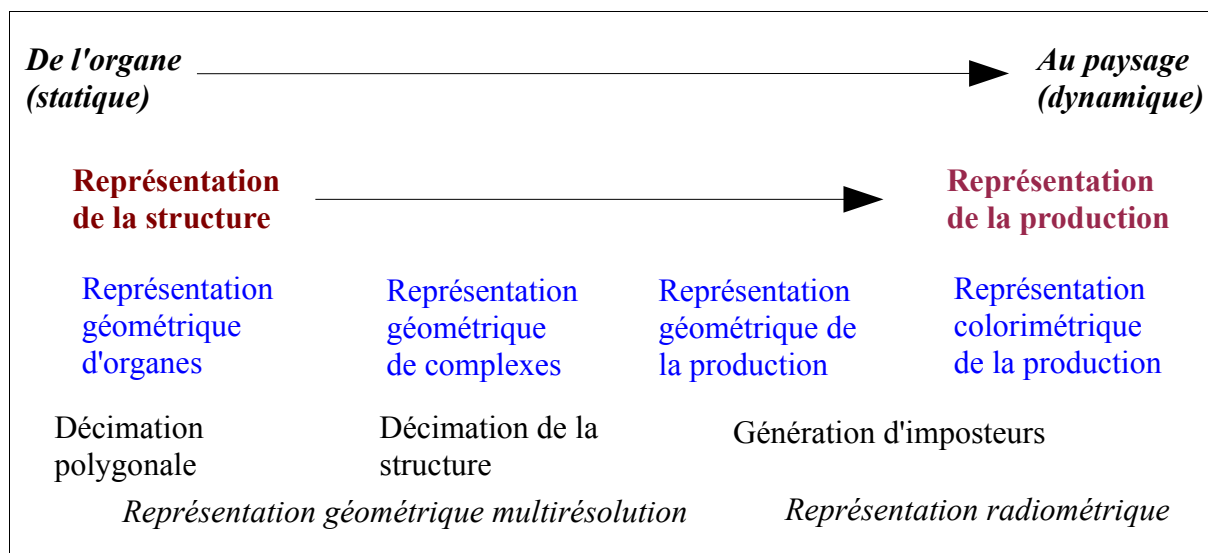


Figure Int_2. Les niveaux de représentation pour la visualisation des plantes et paysage (2003).

Les thématiques de recherche se positionnent à cheval sur ces niveaux, afin de permettre le passage d'un niveau de représentation à un autre. Elles couvrent ainsi:

- la définition de méthodes et approches multi-échelles décrivant la structure de la plante
- la définition de géométries attachés au fonctionnement de la plante et leurs représentations
- la simulation de la dynamique des paysages et sa visualisation

Ces thèmes définissent un canevas sur le long terme, bien sur ambitieux, et abordé que très partiellement ici. La stratégie s'appuie sur le développement d'interfaces appropriées entre les différents niveaux de détail, en respect avec les approches et outils existants, assumant une compatibilité ascendante.

Les modèles multi-échelles géométriques

Notons que l'on ne se pose pas ici la question de la génération des plantes dite virtuelles; elles peuvent être simulées (dans notre cas par une implémentation du modèle GreenLab) ou

reconstruites.

La définition de modèles géométriques multi-échelles (modèles LoD) est certainement le thème sur lequel nous avons le plus avancé à ce jour. Un chapitre y sera consacré plus en détail.

Il s'agit de créer des modèles multi-échelles par agrégation/décimation de la structure

selon des contraintes topologiques (liés à la topologie de la plante)

selon des contraintes géographiques : (liées à la proximité, exemple cluster de feuilles)

selon des contraintes fonctionnelles (peu exploré à cette heure)

Les modèles géométriques représentant le fonctionnement

A ce jour, nous avons conduit fort peu de travaux dans ce sens. Ils sont aujourd'hui limités à la représentation de la biomasse cumulée, soit plongée selon le squelette des plantes dans un espace voxel, soit plongée sous une forme simple (sphère). Dans la suite de l'exposé, on ne consacrera pas de chapitre à ce sujet mais on inclura ces deux volets respectifs dans le chapitre consacré aux représentations LoD, dans le chapitre concernant les paysages dits fonctionnels. Dans le dernier chapitre, nous donnerons quelques lignes de développement que nous prévoyons de conduire .

Les paysages dits fonctionnels

On désigne sous ce terme abusif une imagerie destinée à représenter des dynamiques au sein de paysages virtuels. Le terme de fonctionnel est utilisé en référence à l'imagerie médicale, pionnière dans le domaine. Des avancées intéressantes ont été menées sur ce thème, en s'impliquant bien en amont de la représentation, sur la simulation de la dynamique elle-même ce qui sera décrit dans le dernier chapitre.

Le découpage présenté peut également être perçu comme un axe de complexité ou d'intégration de la représentation explicite de la structure des plantes vers la représentation des sorties fonctionnelles d'un paysage. Selon cet axe de différenciation, la définition explicite de la géométrie, que l'on peut abusivement qualifier de vectorielle, s'appuyant sur les composants locaux à représenter, devient de plus en plus agrégée pour aboutir à une représentation de l'espace et non plus des éléments le composant, mais dont on représente une valeur fonctionnelle.

La définition de paysages selon des grilles est certes limitative mais permet d'exploiter des entrées et des méthodologies éprouvées. Elle assure un pont vers les systèmes d'information géographiques, et le monde de la cartographie en général [ST_Geoma010]. Elle est aussi naturelle dans le contexte de la synthèse d'image et d'outils populaires dans la génération d'image de paysages virtuels [ST_Terra010], [ST_WCS010], y compris pour les forestiers [ST_Forest010].

Cette stratégie tente enfin à contribuer à la définition de paysage dynamiques; l'on emploie ici abusivement le terme de paysage fonctionnel; vu comme des systèmes complexes, consistants et cohérents.

CHAPITRE I

La représentation de la structure des végétaux

et de structures anatomiques

La modélisation et la simulation de la géométrie des plantes simulées

Dès l'aube du développement de la synthèse d'image, on a vu apparaître de multiples contributions visant à calculer et représenter des formes naturelles, et en particulier des plantes.

Plantes virtuelles: quelques approches

Sans rentrer dans un historique des modèles de création de plantes virtuelles, on peut aujourd'hui classer les approches selon 3 grandes modalités: constructive par la génération procédurale de la structure arborée en s'appuyant sur des règles plus ou moins relative à une croissance; descriptive par la définition géométrique d'une enveloppe; reconstruite en 3D, à partir de données réelles acquises par un système automatisé ou non. Cette classification est caricaturale, et il est courant de voir des approches hybrides. Il n'est pas dans l'esprit de ce document de faire un état de l'art exhaustif de ces approches, on peut pour cela se rapprocher d'ouvrages tels que celui de O. Deussen et B. Lintermann [MP_Deussen004] ou P. Prunsinciewitz [MP_Prunsin990], ou des cours (anciens) du Siggraph (Siggraph Courses on Natural Phenomena -1987, 1993, -1996, ...). Nous nous contenterons ici de rappeler quelques travaux dans le but de positionner l'approche « AMAP ».

Les approches par règles sont certainement les plus usitées pour générer des plantes virtuelles. Depuis de simples récursions inspirées des fractales [MP_Kawagu982], [MP_Oppen986], la communauté de la synthèse d'image a exploré de nouvelles approches. Basées sur la loi de gravité -les systèmes de particules [CG_Reeves985], basées sur les modèles utilisés en hydrologie [MP_Eyroll986], et bien sur inspirée de la biologie [MP_Linden968] avec un développement conséquent autour des grammaires L_systèmes [MP_Prunsin988], [MP_Prunsin990]. Il n'y a pas de cloisonnement entre ces approches, en fait on trouve une multitude d'approches procédurales plus ou moins spécifiques s'appuyant sur des règles qui reflète la mise en place de la ramification, sans être forcément en liaison avec la croissance au sens biologique du terme. On peut par exemple citer l'approche d'Aono et Kunii [MP_Aono984], les graftals [MP_Smith984], ou des combinaisons de "patterns" [MP_Weber995], [MPU_Deussen996].

Les approches géométriques "globales" ont été peu explorées, bien qu'abondamment employées dans le milieu de la recherche forestière [ST_Forester010], [ST_SmartForest000], [ST_WCS010]. La raison est principalement liée au fait que l'on se limite à la caractérisation d'une couronne, supportant mal des vues rapprochées. En synthèse d'image, les représentations sous formes d'ellipsoïdes sont parmi les plus anciennes [MP_Gardne984].

La reconstruction tridimensionnelle est en plein essor aujourd'hui. Elle s'appuie sur des modalités différentes, de la simple image à l'exploitation de plusieurs jeux d'acquisition laser. En général le traitement des acquisitions permet de construire soit des squelettes soit des cibles qui vont définir la structure ramifiée [MRP_Quan006], [MRP_Reche004], [MRP_ShlyakT001], [MRP_Tan007], [MRP_Tan008]. Cette dernière est affinée par des approches utilisant localement des règles (empiriques, systèmes de particules, grammaires) pour garantir la continuité des axes et générer les bout d'axes ou les axes périphériques. Des procédures spécifiques, inspirées d'approches semblables, sont ensuite employées pour générer le feuillage [MRP_Neuber007]. Les reconstructions tridimensionnelles à partir d'échantillonnages ponctuels [MPU_Bloom985] sont de plus en plus développées avec l'essor des acquisitions laser [MRP_Chen007], [MRP_Xu007], [MRP_Ning010].

Il existe aujourd'hui de plus en plus d'interactions et de ponts entre ces diverses approches. Lorsque l'on considère en particulier les travaux visant à construire des structures réalistes depuis la saisie interactive de silhouette (sketch based generation), on met en œuvre des techniques où l'enveloppe est contrôlée par l'utilisateur; ce dernier choisit ensuite des heuristiques pour générer la structure en définissant quelques traits de la structure et des règles [MPU_Chen008], [MPU_Wither009], ou encore en spécifiant des composants modèles (patterns) [MPU_Deussen997], [MPU_Linter999].

Les modèles Amap et GreenLab

Dans les sciences agronomiques, la représentation de la structure des plantes est une préoccupation plutôt récente et marginale, les efforts de modélisations, initiés avec le développement de la biométrie il y a plus d'un demi-siècle, portent surtout sur le fonctionnement et la production, indépendamment de la structure et surtout indépendamment de la géométrie de la structure.

La simulation de la structure topologique

L'approche architecturale issue de la botanique [MP_Halle978] va permettre le développement des premières simulations de plantes "réalistes". C'est le cas du modèle AMAP. L'établissement de la structure s'opère selon des axes construits par des méristèmes dont les propriétés sont distinctes, suivant un gradient morpho-génétique (l'âge physiologique). Les axes se développent selon une stratégie spécifique appelé modèle architectural [MP_Barthelemy007]. Dans les stades matures, les instances de cette stratégie (le diagramme architectural) sont dupliquées créant des répétitions [MP_Oldema974]. Philippe de Reffye [MP_deReffye985] va mettre au point la méthodologie de mesure permettant de reconstruire le diagramme architectural et montrer que l'on peut, en respectant la typologie des axes, simuler non seulement la structure mais aussi sa croissance à partir de lois de dormance, mortalité et ramification portant des bourgeons sur les unités de croissance. Cette démarche impliquant biologie, mathématique et informatique (AMAP) permet une génération rapide d'a priori toute plante; dès la fin des années 80, elle suscite l'intérêt en imagerie [MP_deReffye988a].

Dans l'approche AMAP, les règles permettant la création de la structure sont implicites. Nous effectuons la construction de la structure à la volée, en ordre préfixé (pour minimiser les coûts de mémoire), pour un âge donné, en fonction de tirages aléatoires portant sur l'évolution des bourgeons terminaux et latéraux [MP_Jaeger985].

La simulation de la structure géométrique

La sortie géométrique est calculée dans la foulée mais reste de haut niveau: nous estimons les grandeurs des portions d'axes (longueur et rayon), puis calculons les orientations et positions de ces dernières en fonction des angles d'insertion, de phylotaxie et d'un ploieement mécanique simplifié [MP_deReffye989]. D'un point de vue technique nous différencions le premier nœud d'un axe, pour lequel il faut calculer l'orientation de l'axe en fonction de l'axe porteur et évaluer la flèche, des autres nœuds pour lesquels nous appliquons des déviations locales (pour simuler les effets mécaniques en pratique). Les organes sont traités comme un nouvel axe. Pour tout élément de la plante sont donc évalués des facteurs d'échelles, les orientations principale et normale et une origine (absolue ici). Ce qui revient, en fait, à définir un opérateur affine de transformation/positionnement

dans l'espace pour chaque composant de la structure. L'ensemble des opérateurs (constituants) est enregistré sur disque, avec quelques données typologiques telles que la nature du constituant, son âge [MP_Jaeger992]. Cette représentation, que nous avons baptisé Ligne élastique (Line Tree) ne comporte donc pas de topologie mais reste de haut niveau car hautement paramétrable et portable.

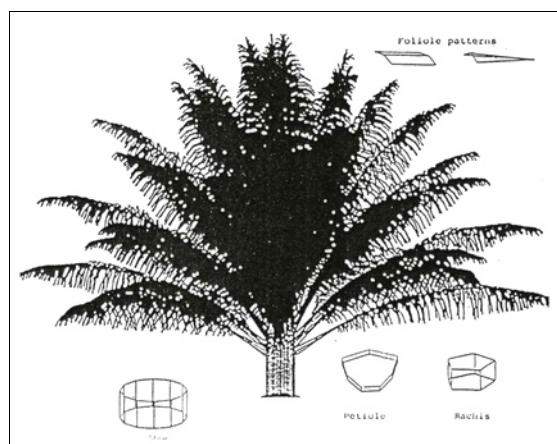


Figure Mod_1. Palmier avec ses formes pour l'habillage (issu de [MP_Jaeger992])

La visualisation d'une plante simulée nécessite donc la définition d'objets graphiques associés à chaque type de constituants (Figure Mod_1). Nous construisons ces formes selon une définition normalisée: même axe principal, inscrits dans un volume unitaire. Leurs géométries sont définies par des approches classiques de synthèse d'image [CG_Bechmann007] dans des éditeurs géométriques dédiés.

En fonction des besoins et des applications, nous obtenons des représentations variées en exploitant des formes différenciées (feuilles dites gauches et dites droites) à géométrie complexe [MP_Lecoustre992] pour un même constituant, ou, à l'inverse, nous pouvons ignorer des constituants pour gagner en vitesse d'affichage (voir chapitre suivant) ou obtenir des effets saisonniers (arbre en hiver par exemple).

La représentation de la géométrie dans les modèles GreenLab

Dans le modèle GreenLab, on modélise et simule non seulement la structure mais aussi le fonctionnement (la production). Nous ne détaillerons pas ici le modèle, dont les principes sont exposés dans [MP_Courne005] [MP_deReffye008a], et détaillés dans [MP_deReffye003a]. Mais nous sommes conduits à expliciter des différences avec AMAP, relatives aux aspects géométriques. Dans l'approche GreenLab, les règles d'établissement de la structure sont explicites, implémentées par un automate double-échelle [MP_Zhao003] ou encore une grammaire [MP_deReffye003b]. Mais la structure n'est pas explicite: le modèle nécessite de connaître le nombre d'organes formés par unité de temps et par âge de différenciation. Les simulateurs génèrent ainsi des représentants des axes et de leurs hiérarchies: les sous-structures [MP_Yan003], elles sont ensuite ré-assemblées dans les simulateurs pour construire la structure (Figure Mod_2), si et seulement si une sortie tridimensionnelle est voulue. La taille des constituants, c'est à dire la longueur, le diamètre des axes, la surface des feuilles, le volume des fruits... sont calculés et non pas estimés, chaque grandeur est une résultante cumulée de la production de la plante. Du point de vue géométrique il y a donc de réelles différences dans l'établissement de la géométrie entre les approches AMAP et GreenLab. Dans GreenLab, la construction de la structure géométrique de la plante est un post-traitement, réalisé par les instanciations des différentes sous-structures constituant la plante simulée.

Ceci étant dit, les sorties tridimensionnelles de simulations pour la visualisation restent, à ce jour les même que celles d'AMAP (la ligne élastique). On dispose cependant de la possibilité d'extraire la topologie du système avec les instances explicites des sous-structures.

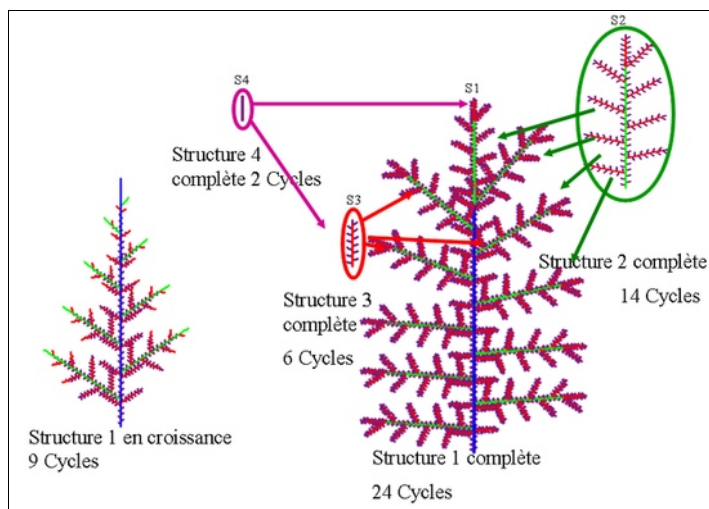


Figure Mod_2. Les sous-structures dans GreenLab (issu de [MP_Yan003])

A ce jour, ces instances sont cependant peu employées pour la visualisation car elles ne peuvent intégrer des déformations liées au positionnement ou à l'orientation et donc ne peuvent rendre compte d'effets mécaniques ou de tropisme géométriques qui contribuent significativement au réalisme des représentations visuelles des plantes simulées.

La visualisation de plantes et de paysages et leurs développements techniques

Les visualisations des plantes simulées par les modèles du type AMAP ou GreenLab obéissent donc au même schéma: il s'agit d'habiller un squelette inorganisé mais comportant des informations typologiques et géométriques d'échelles, de position et d'orientation. Dans les simulations GreenLab, on peut disposer d'informations topologiques et disposer d'une compression par l'usage d'instances de sous-structures identiques, aux ploiements mécaniques près.



Figure Mod_3. Palmeraie sous la brume (collection de l'auteur, 1992)

Les outils de définition de formes pour les organes, d'attribution de matériaux (au sens de la

synthèse d'image) et de visualisation de plantes simulés relèvent d'une ingénierie logicielle simple basée sur une représentation polygonale utilisant les fonctionnalités usuelles de bibliothèques graphiques telle qu'OpenGL.

Pour les paysages et cultures, la technique reste inchangée. On tire cependant profit de la structure de ligne élastique pour construire des représentations multi-échelles (voir chapitre suivant). Les outils de visualisations des scènes végétales capitalisent les acquis sur les développements des modèles multi-échelles et permettent la visualisation de plantes simulées ou reconstruites sur des terrains définis simplement par une grille d'élévation.

Le réalisme est accentué par l'usage de textures, le calcul d'ombrage (approches par shadow maps [CG_Reeves987]) et des fonctions de post-traitement appliquées sur les images munies de leurs tables de profondeurs (zbuffer): brume, pluie, neige, flou de profondeur de champ, effet d'halo (High dynamic Range), flou de profondeur de champ, ... (Figure Mod_3 et Figure Mod_4).



Figure Mod_4. Scène avec Post-traitements (collection de l'auteur, 2006)

Cependant, depuis 2001, d'autres voies sont visitées, mais qui, pour l'heure, ne sont pas encore intégrées dans les outils destinés aux applications auprès de partenaires. Il s'agit de travaux concernant l'illumination et plus particulièrement l'ombrage dans les scènes complexes, et d'explorer le potentiel qu'apportent les sous-structures avec des rendus par points (point splatting). Ces travaux ont été effectués au LIAMA sous notre co-encadrement à l'UMR Amap. Trois voies complémentaires furent et sont explorées.

Le rendu de scènes à éclairage complexe.

Le rendu réaliste de scènes complexes soumises à des éclairages complexes pour des vues du type intérieur de cathédrale, ou sous couverts de végétation dense (canopées) est généralement abordé par des techniques de radiosité ou de lancer de rayon employant des cartes d'environnement lumineux [CG_Debev998]. Notre contribution avec le doctorant Mei Xing, réside dans l'échantillonnage de la carte sphérique d'illumination, découpée en régions équivalentes en terme de distribution [CG_XM006]. Ce découpage permet d'optimiser les calculs de rayons réfléchis dans les approches de lancer de rayons distribués. A ce jour, l'approche reste lourde sur les scènes végétales, elle intéresse en fait plutôt nos partenaires agronomes spécialisés sur les transferts radiatifs que les

usagers des visualisations.

Le calcul de scènes avec une illumination globale.

Le calcul d'ombrage avec une approche dite d'occlusion ambiante (*ambient occlusion in english*) apporte un réalisme spectaculaire mais présente une grande complexité algorithmique car devant calculer les inter occlusion des éléments de la scène [CG_Lang000]. L'utilisation d'un tel modèle d'illumination, global, paraît illusoire sur les scènes végétales.

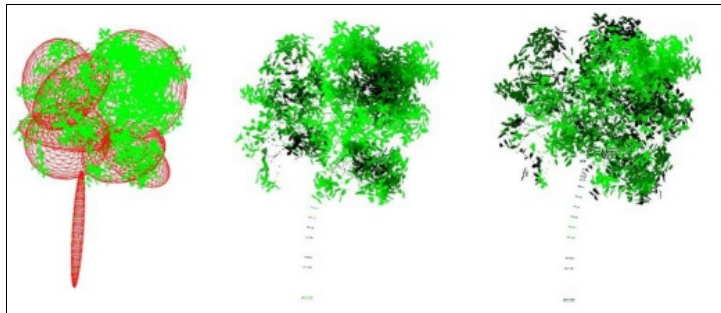


Figure Mod_5 Ellipsoïdes statistiques, valeurs d'occlusion internes, et valeurs cumulées internes et externes (issu de [CG_JT007b])

Avec le doctorant Teng Jun au LIAMA, nous montrons pourtant qu'en découpant les couronnes végétales en clusters, et en utilisant une approche statistique à deux niveaux on peut définir une illumination ambiante performante et réaliste [CG_JT007a]. Pour chaque sous couronne, nous calculons une valeur d'occlusion dite interne, fonction de la profondeur dans l'ellipsoïde statistique obtenu par l'analyse des dispersion des positions des organes. Dans un deuxième temps on calcule un facteur d'occlusion dit externe entre les ellipsoïdes caractérisant les sous couronnes (Figure Mod_4).

Les deux facteurs d'occlusion sont alors combinés pour fournir un coefficient d'atténuation des contribution diffuses et spéculaires de l'illumination [CG_JT007a], permettant ainsi l'obtention d'une représentation réaliste avec des ombrages doux, typiques de ce type d'approche (Figure Mod_5).



Figure Mod_6 Rendu sans et avec occlusion ambiante (issu de [CG_JT007b])

Ce travail, non poursuivi à ce jour, gagnerait à être exploité dans le court et moyen terme. En effet, la définition des sous couronnes, laissée à l'utilisateur est aujourd'hui totalement automatisable en s'appuyant sur les sous-structures. Par ailleurs, le calcul des occlusions internes n'est pas sensible à l'orientation et au positionnement, et il peut se définir récursivement en fonction des sous structures incluses; on peut donc tirer profit de l'instanciation de ces dernières. C'est donc un sujet intéressant

à revisiter.

La représentation des sous structures par fonctions implicites (branches)

Les représentations construites sur la base de surfaces implicites possède de bonnes propriétés esthétiques. J. Bloomenthal avait déjà exploité ces propriétés pour représenter la base de troncs [MPU_Bloom986]. Avec le développement d'approches incrémentales [CG_Bloom990], on peut envisager des représentations implicites [CG_Sclaroff991], en restant dans un contexte de rendu par point (point splatting) adaptées à la structure végétale [CG_Figue996]. L'idée, que nous avons explorée avec J. Teng et W. Yi au LIAMA est d'échantillonner la surface implicite [CG_Witki994] selon une construction incrémentale. La représentation est initialisé sur chaque axe, indépendamment les uns des autres. Puis nous fusionnons les axes lorsque cela est nécessaire. La ramification est traitée en deux étapes : élimination puis migration des points dans une boîte englobant la zone de branchement [CG_Yin005] (Figure Mod_6).

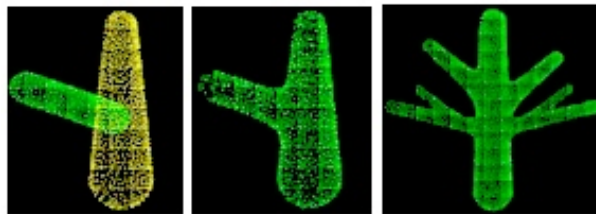


Figure Mod_7 Reconstruction implicite à partir de squelette. Les représentations des axes sont fusionnés entre eux. L'approche exploite les instance de sous-structures (issu de [CG_Yin005])

L'intérêt de cette approche est triple, on est en présence d'une approche hiérarchique, constructive, elle exploite bien les instanciations des sous-structures et, enfin, elle montre une résolution propre des ramifications, ce qui suscite un intérêt en qualité des bois pour des représentation de simulation de découpes.

Ce travail, non poursuivit à ce jour, gagnerait également à être revisité dans le court et moyen terme. C'est l'un des premier travail de représentation exploitant le cadre des sous-structure. Il n'a pas pu à l'époque s'exprimer totalement faute de données et de maturité des outils de simulation. Il en est tout autrement aujourd'hui. Par ailleurs la représentation de fonctions implicites dont les primitives sont limités à des axes a grandement évolué avec l'usage des fonctions de filtrage dans les cartes graphiques (GPU). On peut clairement envisager d'écrire des shaders générant la surface par une méthode de splatting, sur la seule donnée de portions d'axes définis par une liste de nœuds valués de leur diamètres. Cette approche, combinée au résultats obtenus sur les représentations multi-échelles et sur la génération de la géométrie à partir de méta-données (voir chapitre suivant) est clairement à revisiter dans un contexte de génération de primitives géométriques par la GPU.

Les reconstructions géométriques de plantes et structures anatomiques

Dans le domaine médical, tout comme dans d'autres domaines, l'accès à la représentation géométrique d'un organe (ou objet naturel) réel et précis n'est pas trivial, il se doit d'être non destructif et doit refléter non pas une conceptualisation mais une représentation de l'instance réelle, avec ses pathologies structurelles éventuelles. Le recours à des systèmes d'acquisitions est donc inévitable. Ces derniers sont variés et de plus en plus populaires : le cliché numérique, les séries d'images, les volumes d'images (X, nucléaire, PET,...), les acquisitions laser...

L'extraction de la géométrie n'est pas un simple processus, il est dépendant des systèmes d'acquisitions, de la nature de l'objet, et, s'inscrit en fin d'une chaîne de traitements longs et complexes : segmentation / classification / caractérisation / reconnaissance / reconstruction.

De plus, la majorité des acquisitions ne permettent qu'une acquisition partielle de l'objet d'intérêt, bruitée, présentant des occlusions et des artéfacts. L'objet d'étude est par ailleurs rarement isolé, il faut l'extraire de son contexte. Et arrivé à ce stade, il n'est pas acquis que l'on puisse isoler ses constituants et reconstruire des niveaux hiérarchiques, ce qui est important pour des systèmes ramifiés.

Des structures ramifiées végétales à l'imagerie médicale ?

L'expérience acquise sur la caractérisation des structures ramifiées végétales, leurs modélisations, simulations et représentations tridimensionnelles nous ont conduit à des partenariats inédits avec le milieu médical; avec comme objectifs d'appliquer cette expérience aux arbres hépatiques puis à la caractérisation et à la modélisation géométrique d'organes en général. Les difficultés d'acquisition de données ont rapidement centré les travaux sur l'exploitation de l'imagerie médicale, avec des objectifs modestes. Ces développements se sont effectués de 1990 à 2000 sur la base d'acquisitions tomodensitométriques (scanner RX).

Cette expérience fut d'un grand intérêt. Nous ne détaillerons pas ici l'ensemble des travaux conduits dans cette période mais soulignerons des points particuliers, points qui ont permis l'évolution de la réflexion sur les applications concernant la représentation des plantes et l'émergence de ce que nous appelons les "paysages fonctionnels". Ces points concernent la visualisation d'images volumiques, la segmentation et la caractérisation statistique de la géométrie des organes. Nous listons quelques applications médicales avant d'aborder les thématiques de reconstructions géométriques de plantes menées aujourd'hui en collaboration avec le LIAMA et le Lrfob, à partir de données laser.

De la visualisation à la caractérisation avec l'imagerie volumique

Visualisation.

Les acquisitions d'images volumique présentent l'intérêt de permettre une grande variabilité d'approches de visualisation de leur contenu.

Dans nos applications avec les partenaires médicaux ou industriels, nous sommes généralement en présence de deux espaces, l'un de labels, issus d'un processus plus ou moins long de traitement, caractérisant la nature du contenu des voxels, l'autre d'un champ scalaire, caractéristique du système d'acquisition (la densité tomodensitométrique pour le scanner X). Nous avons implémenté des visualisations "classiques" de ces espaces [IV_Cline988], [IV_Lorens987], [IV_Kaufm990], avec

quelques originalités concernant les images labels [CG_Borianne993]. Mais les originalités des travaux trouvent leurs natures dans l'application de la combinaison de ces techniques dans un contexte de recherche clinique [IV_Briand993a], [IV_Briand993b], [IV_Jaeger993].

En fait, en visualisant en plusieurs passes les 2 espaces avec des techniques distinctes (dividing cube, lancer de rayon avec fonction de transfert, rendu au trait), et en intégrant des constructions géométriques issues de caractérisations, l'accès à des acquisitions inédites a surtout permis des visualisations nouvelles dans des contextes non familiers en leurs temps: en embryologie [IV_Christol996], en odontologie [IV_Treil994], [IV_Treil997] (Figure Vol_1).

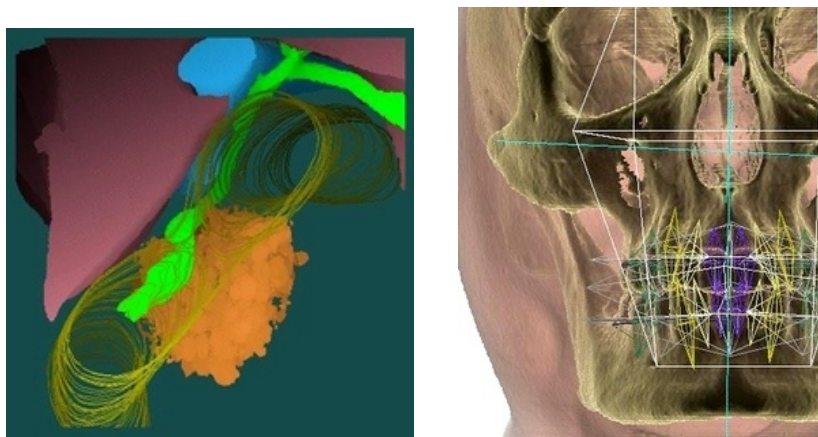


Figure Vol_1. Visualisation volumique (combinaison d'approches) en embryologie (issu de [IV_Christol996]), et en odontologie (issu de [IV_Treil997])

Ce fut également le cas dans des domaines non médicaux tels qu'en foresterie [IV_Jaeger999], en paléontologie [IV_Jin007], et en agronomie-agroalimentaire [IV_Nicolini992] (Figure Vol_2)

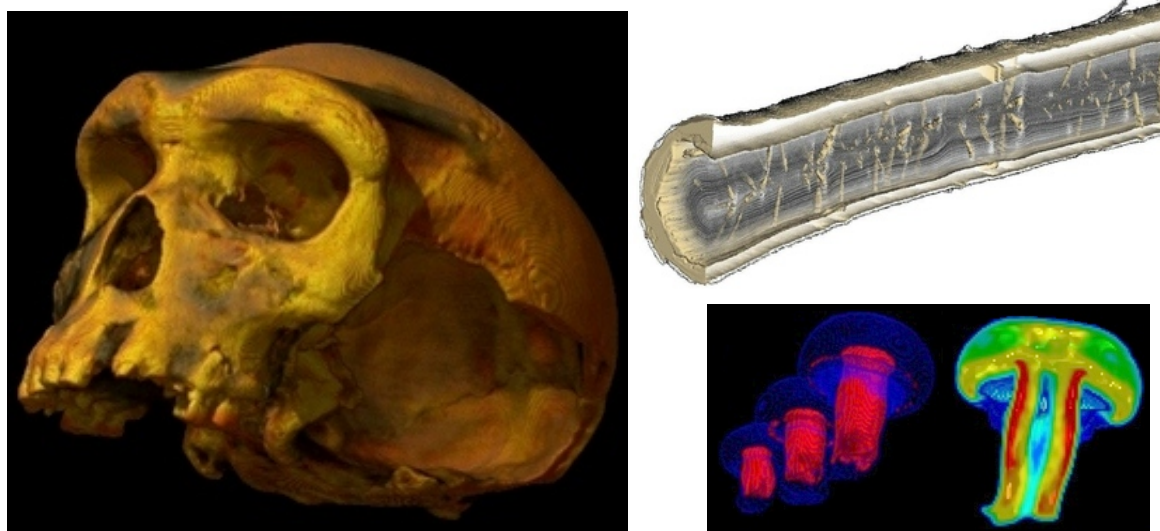


Figure Vol_2. Visualisations volumiques par fonctions de transfert en paléontologie (2002), en foresterie (issu de [IV_Jaeger999]), et en agronomie (issu de [IV_Nicolini992]).

Les fonctions de transfert [IV_Pfiste001] permettent d'associer couleurs et transparences aux valeurs scalaires des voxels. Elles contribuent ainsi à visualiser des propriétés des structures lorsque les valeurs traduisent des grandeurs de fonctionnement. Ainsi, sur le champignon de Paris, nous

avons montré que la réponse aux rayons X était liée aux taux de divisions cellulaires [IV_Nicolini992] (les zones de croissance en couleurs chaudes dans la figure Vol_2).

Segmentation et caractérisation de structures anatomiques.

Les outils d'identification de structures anatomiques font appel à des fonctions de traitement d'images classiques (seuillages, détecteurs de contours [TIS_Derich987], etc...) Pour certaines applications, ces outils étaient insuffisants, nous avons développé avec S. Chemouny en thèse un ensemble de filtres non linéaires tridimensionnels [IV_Chemouny999], extensions des filtres de voisinage de Nagao [TIS_Nagao979]. Ces derniers ont permis de segmenter des tissus pathologiques hépatiques et ont également été testé avec succès sur la détection de branches dans des billons scannés [IV_Jaeger999]. Plus tard, au LIAMA, et plus spécifiquement en partenariat avec le NLPR dans le cadre de la thèse de ZhiFeng Cai, nous avons adapté des modèles de traitement d'image de segmentation employant des techniques de mise en correspondance s'appuyant sur des modèles de déformation visco-élastiques [TIS_Cai002], [TIS_CAI003]. Plus récemment, nous avons employé les outils dans des applications en biologie, pour l'exploration des structures racinaires reconstruites à partir de coupes sériées colorées microscopiques [IV_Wu010].

La caractérisation des structures géométriques identifiées s'appuie sur l'analyse statistique de la distribution du nuage de points (les centres des voxels) du solide étudié. Le calcul des composantes principales (en fait les axes d'inerties ici) est employé pour définir la direction principale des organes étudiés (voir Figure Vol_1 sur les dents [IV_Treil997]). Ce calcul fut utilisé dans plusieurs études cliniques; pour détecter des pathologies liées à la croissance osseuse [IV_Canovas000], des ruptures ligamentaires, ou encore employées comme signatures d'os en orthopédie [IV_Bonnel000].

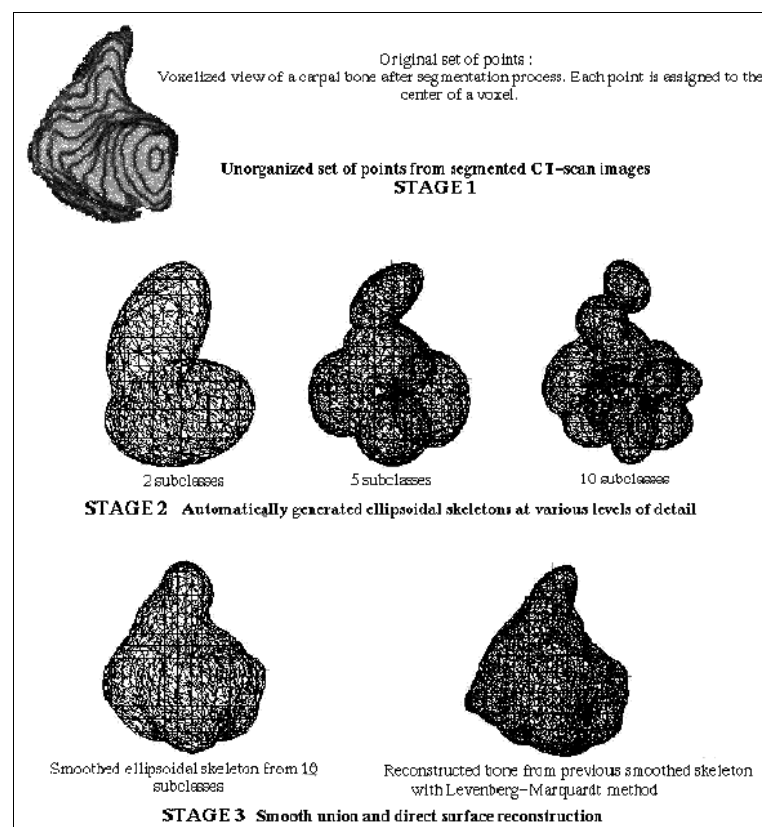


Figure Vol_3. La caractérisation d'un solide (l'os pyramidal du carpe) par son E-squelette et sa reconstruction (issu de [IV_Banegas999a]).

Nous avons conduit un travail plus fin dans le cadre de la thèse de Frédéric Banégas. Nous décomposons hiérarchiquement le nuage de points (voxels) issu de la segmentation d'un organe par une hiérarchie d'ellipsoïdes en employant le formalisme des nuées dynamiques [TIS_Diday971]. L'ellipsoïde initial est défini par les composantes principales de l'ensemble des voxels constituant l'objet d'étude. Chaque ellipsoïde statistique, est récursivement divisé en deux si sa dispersion est significative. L'intérêt de l'approche est qu'elle présente une décomposition hiérarchique du solide en ellipsoïdes. Cette décomposition en "niveau de détails" nommé E_squelette, est une caractéristique de l'objet, utilisable comme une signature [TIS_Banegas999]. De plus, en attribuant des transformations localement linéaires aux ellipsoïdes – les transformations de Barr [CG_Barr084], nous obtenons une représentation synthétique à partir des ellipsoïdes feuilles (Figure Vol_3) [IV_Banegas999a]. Une peau implicite couvrant les ellipsoïdes transformés permet, in fine, d'obtenir une représentation réaliste de l'enveloppe du solide reconstruit [CG_Banegas999].

Cette approche fut également testée sur des plantes simulées, en considérant le nuage de points formé par les positions des centres des feuilles [IV_Banegas001] (Figure Vol_4).

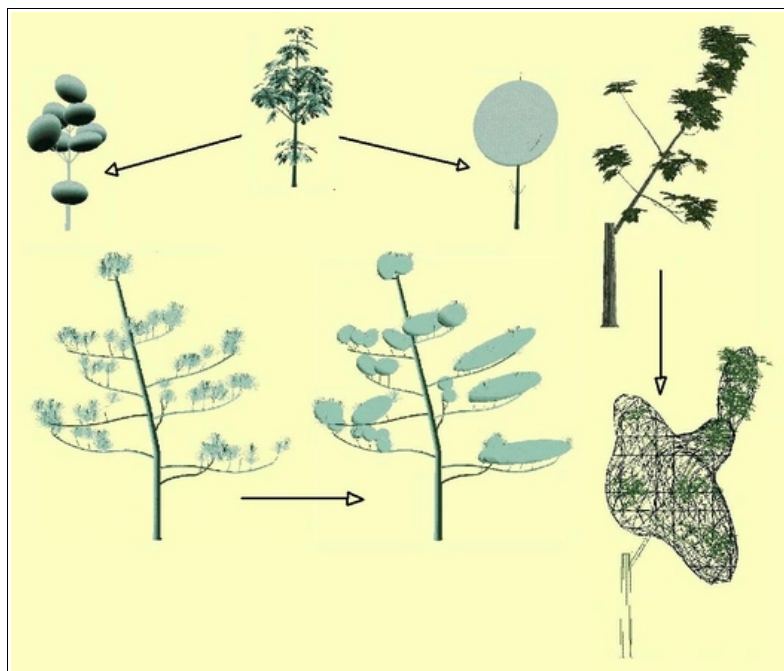


Figure Vol_4. Applications des E-squelettes aux plantes virtuelles (collection de l'auteur, 1999).

L'objectif de ce test était de construire des surfaces de couronnes faciles à intersecter avec des lancers de rayons pour effectuer des calculs de transferts radiatifs. Cette approche statistique est en sommeil depuis 2001, mais devrait être revisitée sur les acquisitions laser de scènes forestières en particulier. L'analyse de structures ramifiées, l'une des motivations initiale des projets médicaux, fut en fait abordée bien plus tard avec le LIAMA

La décomposition hiérarchique d'objets ramifiés dans un espace voxel

Nous avons mené cette analyse en coopération avec l'équipe de Zhang Xiaopeng au LIAMA. Elle vise à reconstruire la structure d'un objet ramifié plongé dans un espace voxel.

Le procédé opère en plusieurs étapes. Une classification des voxels permet de mettre en évidence des graines (les bouts d'axes). On classe les voxels en fonction des distances aux graines. Un

premier graphe est construit par adjacence des voxels et évolution des distances. Des branches (à distance constante) on remonte aux ramifications, découpant ainsi le volume voxel. La ramification est ensuite étudiée localement et le graphe revu en conséquence [TIS_Zhang009]. Enfin, le graphe est optimisé et valué pour construire un squelette hiérarchisé.

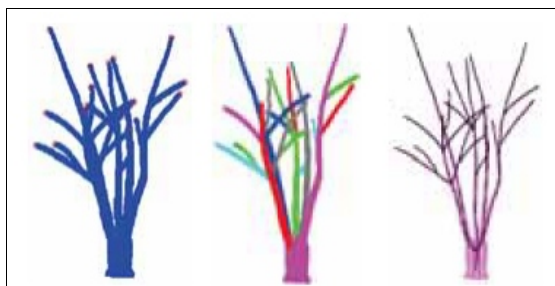


Figure Vol_5. Données initiales, découpage des volume et construction du squelette hiérarchique d'un arbre plongé dans un espace voxel (issu de [TIS_Zhang010]).

L'approche est explorée sur des acquisitions scanner d'arbres bronchiques et sur des plongements de plantes simulées dans l'espace voxel (Figure Vol_5). Ces travaux ont donné lieu à la rédaction d'un chapitre d'ouvrage dans Pattern Recognitions [TIS_Zhang010].

Les apports de l'imagerie volumique pour la représentation des plantes

Les développements en imagerie volumique pour les applications médicales sont clos depuis 2001. La valorisation des travaux, qui a permis le développement de 2 start-up reste toujours en cours. De multiples contrats industriels viennent en effet au terme de clauses de secret.

Dans l'orientation des axes de recherche, cette incursion dans l'imagerie volumique et les applications médicales fut d'une grande richesse et change notre perception de la représentation du monde végétal.

Les outils et approches développées en imagerie volumique vont en effet former le premier socle pour héberger les représentations fonctionnelles de paysages. En plongeant les squelettes des plantes virtuelles dans l'espace voxel (voir chapitre suivant), on va changer la nature même de la représentation du paysage: celui ci n'est plus construit à partir de constituants individualisés (plantes, sol, etc, ...) mais devient une représentation de l'espace lui même, dont les cellules, homogènes, contiennent un volume de biomasse. On est donc bien en présence d'une représentation fonctionnelle du paysage.

Les fonctions de transfert, les outils d'analyse et de caractérisation sont exploitables sur les représentations discrètes de paysages. Bien qu'ils ne soient pas adaptés à la thématique, ils vont inspirer les outils spécifiques d'exploration des paysages simulés (voir dernier chapitre). On va retrouver des fonctions de définition de palettes de couleurs, de transparences et de "mapping" pour visualiser des champs scalaires, et bien sur le principe de représentations multi-passes fusionnant les représentations. Des applications finalisées du domaine médical nous retiendrons enfin les interfaces conceptuelles et les représentations en "tableau de bord".

Mais l'héritage va plus loin. Dans les applications scanner conduites (sur le bois, les champignons, les ananas), les images Rx révèlent en fait peu d'information sur les structures -assez homogènes-

mais traduisent surtout une contenance en eau. Or, dans le modèle Greenlab, le calcul même des volumes (de la production de biomasse) est une fonction simple de la prise d'eau (la Water Use Efficiency). Il devient donc naturel de franchir un nouveau pas: celui de faire "circuler" de l'eau dans les voxel; l'idée de la simulation du paysage fonctionnel s'impose alors [IV_Jaeger003].

Les développements autour de l'imagerie médicale nous ont aussi permis de nous familiariser avec le traitement d'image en général et la reconstruction 3D thème sur lequel nous collaborons avec notre équipe partenaire associée au LIAMA.

Les modèles de plantes définis par reconstruction 3D.

La reconstruction 3D de plantes à partir d'images simples ou multiples ou d'acquisition laser forme une thématique récente, en plein essor. Aux travaux pionniers de [MRP_ShlyakT001] lourds à mettre en œuvre, ont succédé des approches constructives utilisant des espaces voxels intermédiaires [MRP_Quan006] ou des systèmes de particules [MRP_Neuber007] migrant de l'enveloppe de la couronne vers les axes principaux plus facilement identifiables. Les reconstructions issues de lasers terrestres commencent à intéresser la communauté de la synthèse d'image [MRP_Cheng007], [MRP_Xu007] , [MRP_Ning010].

La reconstruction 3D ne constitue pas pour l'heure une activité de recherche centrale à l'UMR Amap mais elle est importante au LIAMA, dans l'équipe de Zhang Xiaopeng. La collaboration entre les deux sites s'organise selon un découpage thématique, voire disciplinaire. Les aspects, importants, concernant le traitement des données jusqu'à la définition du squelette sont développés au LIAMA. Les aspects relevant de la caractérisation géométrique des couronnes, et de la modélisation de la structure à partir des éléments géométriques extraits impliquent l'UMR Amap. L'encadrement des doctorant est donc partiel sur ce sujet.

La construction des couronnes végétales à partir de points laser.

Dans la thèse de Zhu Cao au LIAMA, l'idée est d'obtenir une surface continue de la couronne d'arbres adultes à partir d'un nuage de point (d'une acquisition laser terrestre). On ne cherche pas ici à définir le convexe mais bien à respecter les concavités sur l'enveloppe du végétal. On espère ainsi également contribuer à compléter la reconstruction de la structure qui est généralement diffuse en périphérie de la couronne.



Figure Rec_1. Points laser et extraction des branches (issu de [MRP_Zhu010]).

La couronne est définie par l'enveloppe d'une triangulation solide [CG_Boisso984]. Cette tétraèdrisation est construite par le formalisme des Alpha shapes [CG_Edelsbr994], une extension des diagrammes de Voronoi. La paramétrisation (Alpha) permet d'assurer des tétraèdrisations paramétrables allant du convexe à l'ensemble des points. Par un processus itératif, nous calculons la valeur alpha pour laquelle l'enveloppe de la tétraèdrisation reste une unique surface (manifold) mais qui peut présenter un genre supérieur à un [MRP_Zhu008]. Cette itération converge forcément du simple fait des propriétés des Alpha Shapes. L'itération est cependant longue et coûteuse de part le nombre de tétraèdres considérés. Dans un deuxième temps, nous avons donc cherché à réduire cette complexité en définissant des clusters au sein des points de la couronne.



Figure Rec_2. Clusters dans la couronne et reconstructions (issu de [MRP_Zhu010]).

Pour ce, on extrait d'abord les branches par propagation d'une fonction géodésique de calcul de distance (Figure Rec_1). Puis on construit des clusters avec les points de couronnes en prolongeant les axes principaux du squelette. On applique enfin le processus de maillage solide par les Alpha shapes et leur optimisation sur chacun des clusters identifiés, réduisant ainsi notablement les temps de calculs [MRP_Zhu010] (Figure Rec_2). L'approche reste cependant à ce jour assez laborieuse et devrait prochainement être comparée à l'approche statistique des E-squelette présentée dans le chapitre précédent.

Les développements en cours autour de la reconstruction 3D laser.

Avec le LIAMA, nous conduisons à l'UMR Amap conduit un partenariat de qualité avec l'INRA Forêt (Lerfob) lui donnant accès à de nouvelles nombreuses acquisitions de laser terrestre en milieu forestier. Ces données, de grande qualité, alimentent les développements menés. Les travaux en cours avec les doctorants du LIAMA, Honjun LI et plus marginalement XiaoJun Ning, proposent des contributions sur deux volets applicatifs: la segmentation non supervisée et sa classification en milieu urbain végétalisé; et la segmentation et la classification d'acquisitions laser en milieu forestier.

Principes

Dans ces applications l'ensemble des développements s'appuient sur le codage des points d'acquisition dans un arbre KD (k-d tree) pour un accès rapide et aisé aux voisinages. Dans cette structure d'arbre kd, on définit de nouvelles grandeurs associées à chaque point (calcul de distance, de direction, de normale). De ces grandeurs géométriques, nous procédons à des segmentations régions. Puis, à partir de ces segmentations définissant des régions de points « homogènes », on

cherche à estimer leurs appartenances à des primitives géométriques simples (plan, cylindre) pour obtenir une première classification. Enfin, cette classification est raffinée en étudiant localement les résidus des estimations.

L'extraction des segment d'axes (branches) dans les images laser d'arbres isolés

Sur les arbres isolés, nous pouvons ainsi extraire le feuillage des branches par l'usage de l'étude de la normale et de la direction [TIS_Dai009]. Nous cherchons ensuite à identifier des portions d'axes et à les modéliser par des cylindres (Figure Rec_3) par une méthode de régression classique.

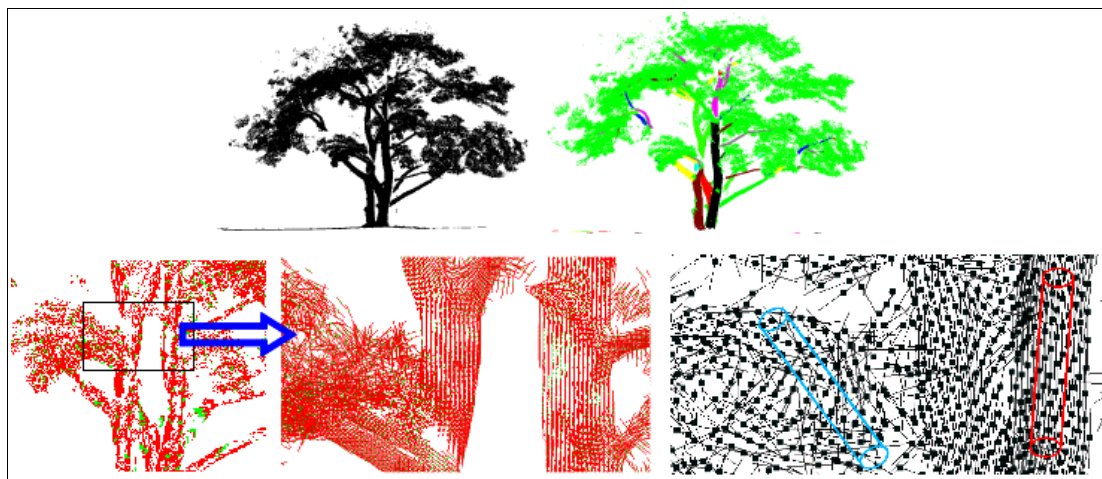


Figure Rec_3. Extraction de branches sur acquisitions laser d'arbres isolés (issu de [TIS_Dai009]). Les points des branches sont extraits selon des critères de direction. On cherche ensuite à estimer localement des cylindres dans ces derniers.

Segmentation et classifications de scènes urbaines

Dans les scènes urbaines végétalisées, la segmentation non supervisée est fondée sur des critères de distance et de normales (Figure Rec_4). Elle permet dans un premier temps de distinguer le végétal des infrastructures. Sur ces dernières, on construit une segmentation région par reconnaissance de plans. La classification est ensuite affinée en prenant en compte de l'importance des surfaces extraites et de l'analyse des résidus locaux.

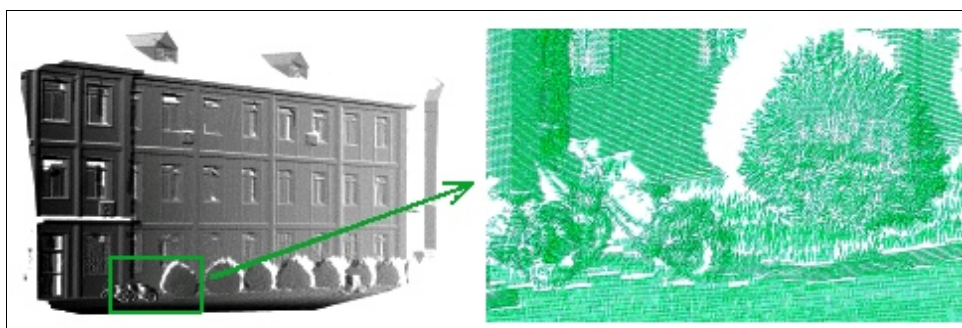


Figure Rec_4. Données laser initiales et calculs des normales (issu de [MRP_Ning009])

Cette approche conduit à une classification hiérarchique [MRP_Ning009] des éléments, elle permet

l'extraction des façades puis des fenêtres (Figure Rec_5). Nous développons depuis peu une classification construite à partir de cette décomposition hiérarchique, en incluant des grandeurs géométriques pour définir des signatures d'objets types, objets courants dans des scènes urbaines végétalisées : arbre d'ornement, buissons, pelouse, parterre fleuri, abri de bus, feux, etc...

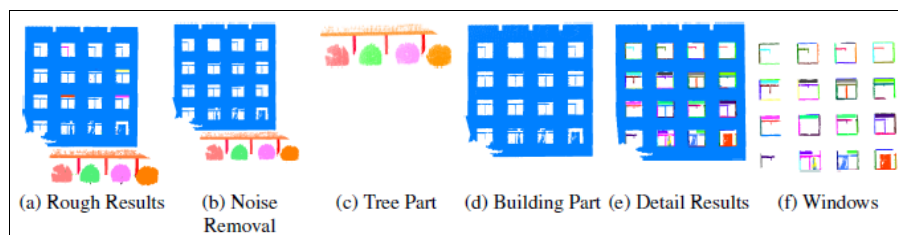


Figure Rec_5. Classification dans une scène urbaine simple (issu de [MRP_Ning009])

La segmentation et la classification d'acquisition laser en milieu forestier.

Dans ce travail développé avec le doctorant LI HongJun, les mêmes outils (distance et normales) sont employés pour détecter des portions de tronc dans les parties inférieures des acquisitions laser. Nous identifions ensuite des sections de cercles dans des coupes horizontales. Ces sections sont filtrées pour reconstruire les troncs. La base des troncs est ensuite utilisée pour affiner le modèle numérique du terrain. Enfin, dans la partie supérieure de l'acquisition, nous attribuons les points des couronnes aux troncs identifiés en prolongeant leurs directions et selon des critères de distance (Fig Rec_6)

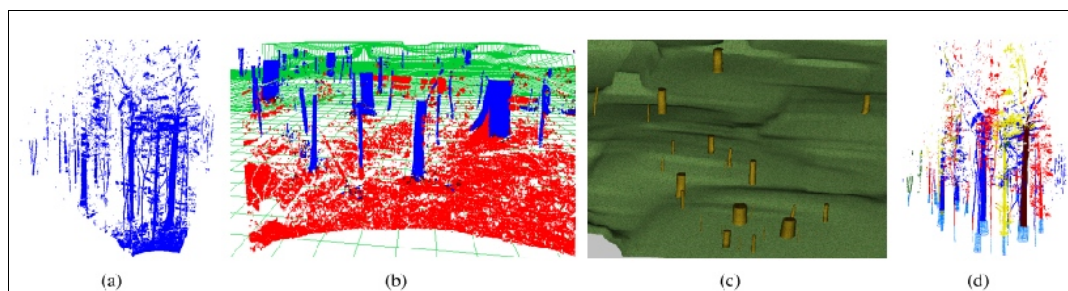


Figure Rec_6. Segmentation d'examen laser forestiers (en soumission, 2010). (a) : données initiales, (b) segmentation de la partie inférieure du scan (c) reconstruction des troncs et du modèle de terrain (d) classification des couronnes.

Plantes simulées ou reconstruites: une même représentation géométrique

Qu'elles soient simulées (AMAP, GreenLab) ou reconstruites, les représentations géométriques des plantes (lorsqu'elle ne sont pas globales) utilisent les mêmes représentations: celles de lignes élastiques composée de constituants inorganisés, et ou celle d'un squelette dont les nœuds porte les valeurs de diamètre et les organes. Les représentations plus "globales" de couronnes par exemple, restent pour l'instant marginales et n'entrent pas à ce jour dans des applications avec des tiers.

Le chapitre suivant s'applique donc insensiblement à l'ensemble des plantes générées, simulées ou reconstruites.

CHAPITRE II

Vers la visualisation réaliste temps réel

des plantes et paysages.

Les problématiques de visualisation temps réel de plantes et paysage abondent dans la littérature. Une typologie sommaire classe les approches selon trois grandes techniques de représentation géométriques: la représentation polygonale, issue d'une définition plus ou moins explicite de la structure [MP_Prunsin990] [MP_Jaeger992], la représentation ponctuelle discrétisant la structure selon des critères définis lors des stades ultimes de visualisation [CG_Gillet005] et enfin les représentations à base d'images, représentant tout ou partie du végétal par une (ou un jeu de) texture(s) plane(s) [CG_Max996], [CG_Jakuli000], [RRB_Behren005], [RRB_Fernan007a]. Une typologie plus fine avec les avantages qualitatifs comparatifs est fournie par Ismael G. Fernandez de l'université de Gérone dans sa thèse de Master [RRB_Fernan007b]. Notre problématique ne se limite cependant pas à une problématique de représentation réaliste de plantes au sens large, mais de visualisation de plantes issues de simulation numériques. Dans notre cadre, on ne peut développer des modèles générant des plantes dans le seul but de la représentation; ce que l'on attend, c'est de visualiser les résultats des calculs. Force est aussi de constater que les résultats des simulations sont lourds, un arbre peut compter des centaines de milliers d'entre-nœud et donc de feuilles.

L'industrie et en particulier la CAO peuvent aussi conduire à des représentations lourdes. L'une des approches traditionnelles aujourd'hui répandue et classique est de générer des géométries adaptées à une contrainte de volume de données en simplifiant la géométrie de l'enveloppe de l'objet. Dans ce contexte, la décimation polygonale est efficace et répandue [LOD_Hoppe996] [LOD_Melax998], elle s'applique bien à la représentation de surfaces continues comme des terrains [LOD_Hoppe998], mais s'apprête mal à la géométrie complexe des végétaux. On a donc vu au cours de la dernière décennie, des approches plus ou moins spécifiques pour ces derniers.

Nous abordons ici dans un premier temps les représentations BREP (polygonales) et ajouterons quelques mots sur les représentations dite images.

La représentation multi-échelle polygonales des plantes simulées

On ne peut pas traiter l'ensemble des constituants du végétal selon une même méthodologie: le bois constitue un système hiérarchique, organisé, devant présenter une continuité géométrique le long des axes, alors que les organes, les feuilles en particulier, portés par cette structure, sont représentés par un ensemble de primitives géométriques éparées, nombreuses avec de multiples occlusions. Toutes les représentations, y compris pionnières font cette distinction, quelles soient procédurales [MP_Aono984], [MP_Eyroll986], [MP_Oppen986], [MP_Prunsin990] ou reconstruites [MPU_Bloom985]. Elle peut parfois être induite par des moteurs impliquant des générateurs spécifiques: grammaires pour la structure, système de particules pour les organes [MP_Smith094].

Modèles et visualisation multi-échelle des systèmes ramifiés (branches)

Dans la plupart des générateurs de plantes, les axes sont formés par une succession de nœuds ou entre-nœuds, autorisant ainsi leurs concaténations.

Nous avons déjà mis à profit cette propriété dans les premiers simulateurs AMAP [MP_Jaeger987], où les entre-nœuds successifs le long d'un axe sont fusionnés; avec des contraintes de nature topologique: il faut que l'axe reste physiologiquement de même nature, et avec des contraintes géométriques: seuls les nœuds successifs respectant une courbure faible sont fusionnés (Figure LoD_01).

De plus, en sortie de simulation, les constituants fusionnés (portions d'axes) sont affectés d'un niveau de détail (1 à 5), en fonction du ratio: diamètre sur plus grand diamètre de la plante. A

l'affichage, selon la distance à l'observateur, ne seront affichés que les constituants estimés importants. Ces outils, bien qu'empiriques furent maintenus dans la chaîne commerciale de BIONATICS [ST_Bionatics010].

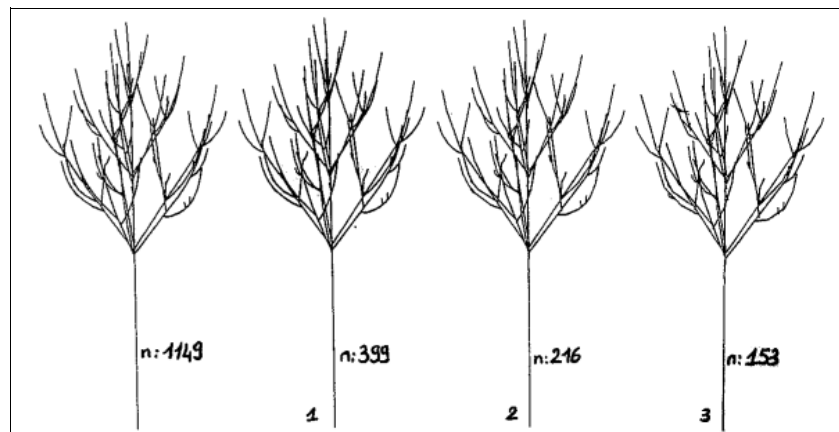


Figure LoD_1. Concaténation des nœuds à l'issue de simulations (issu de [MP_Jaeger987])

La compression et la génération de représentation multi-échelle des structures ramifiées ne fut revisitée que bien plus tard, de 2004 à 2006, au LIAMA, en collaboration avec le Prof. Zhang Xiaopeng. Cette collaboration est toujours active à ce jour.

Le principe de la compression reste semblable. La structure ramifiée est convertie en un squelette dont les nœuds correspondent aux nœuds de la simulation. Le squelette est valué, chaque nœud porte le diamètre de l'axe correspondant. Le squelette est converti en un diagramme de Frenet, de façon à pouvoir quantifier les courbures, et calculer les erreurs de position et direction en chaque nœud lors de fusions. Nous définissons ainsi une série de squelettes à des résolutions diverses, sur lesquels nous pouvons également effectuer des seuillages en fonction des diamètres. Enfin, et c'est ici que réside la principale innovation, nous effectuons le plongement géométrique au moment du rendu, avec un maillage adapté au diamètre de l'axe et à la distance à l'observateur (c'est à dire au diamètre apparent). Si ce dernier est très faible (sa projection inférieure au pixel), l'axe sera représenté par une ligne semi-transparente dont le coefficient d'opacité est fonction de ce diamètre apparent. Ces trois étapes sont illustrées ci dessous (Fig. LoD_2).

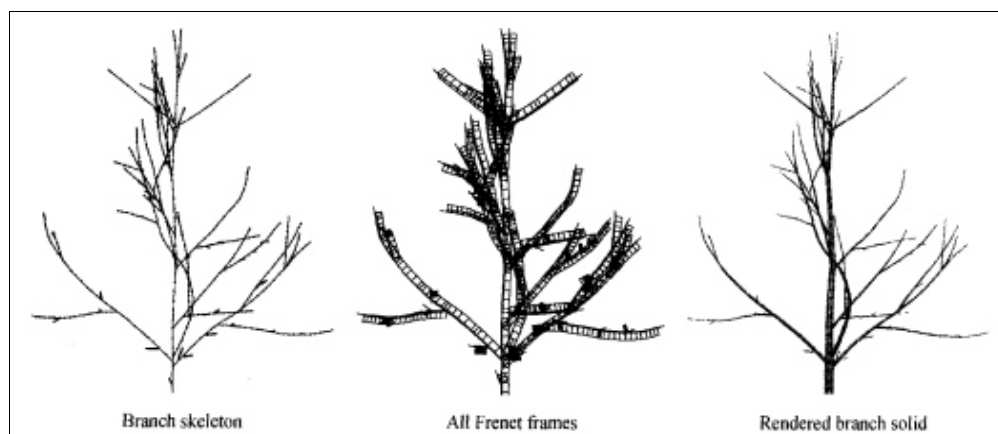


Figure LoD_2. La reconstruction du système ramifié (issu de [LOD_Zhang006b])

Il est à noter que les deux premières étapes sont des prétraitements qui peuvent être intégrés dans les moteurs générateur de plantes simulées, alors que la dernière étape, le maillage, s'effectue lors du rendu, à la volée. Le maillage produit est donc totalement adaptatif comme illustré en figure LoD_3.

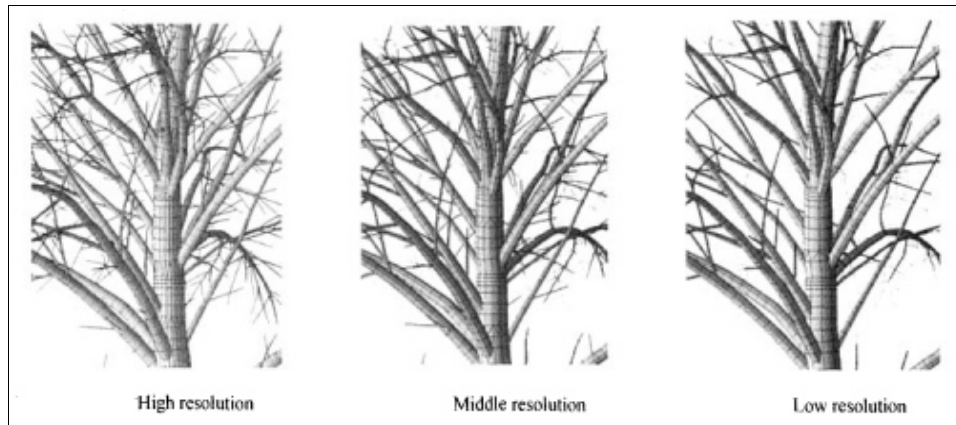


Figure LoD_3. Plongement géométrique multirésolution (issu de [LOD_Zhang006b])

Sur des représentations réalistes, impliquant en particulier du placage de textures, le maillage peut localement être redéfini, de manière à obtenir un placage quasi homomorphe tel que présenté sur la vue réaliste en figure LoD_5.

L'usage combiné de squelettes multi-échelles pré-calculés et du maillage adaptatif généré à la volée permettent une représentation efficace de la structure (figure LoD_4), tout en respectant un grand réalisme dans les vues proches.

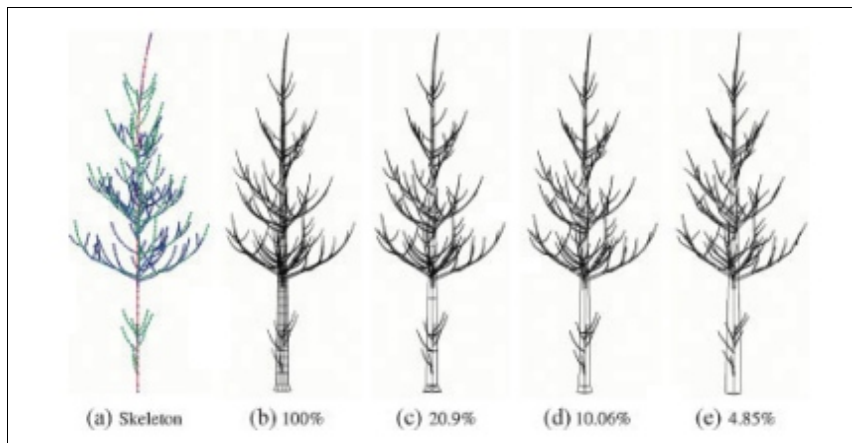


Figure LoD_4. Performance des modèles LOD de branches (issu de [LOD_Deng010])

De plus l'usage de lignes semi-transparentes évite les effets d'aliasing sur les axes fins et apporte un plus tant en performances qu'en qualité graphique (voir ci dessous).

Nous avons implémenté l'algorithmique développée selon plusieurs modalités compatibles avec des environnement de rendus classique, en incluant le placage de texture quasi-conforme. Elle supporte bien des fonctions de rendu et de calcul d'ombrage fondées sur des images de profondeur [CG_Reeves987], bien adaptées aux scènes complexes avec peu de sources (ce qui est notre cas).

Un exemple est présenté en figure LOD_5 (voir aussi les figures suivantes incluant le feuillage).



Figure LoD_5. Vue temps réel dans une peupleraie (collection de l'auteur , 2007)

La représentation et la visualisation multi-échelle des organes (feuilles)

La représentation du feuillage est par nature distincte de celles des branches de part la dispersion des primitives, éparées.

Dans les premiers simulateurs AMAP, les organes ne sont pas soumis à des variations d'échelles. Contrairement aux modèles FSPM tel que le modèle GreenLab, toutes les feuilles d'une plante simulées sont donc identiques et représentées par un même modèle géométrique. De ce fait, dans les anciens travaux, à l'échelle d'une plante, les niveaux d'échelle sont simplement définis par un seuillage aléatoire. Aux 5 niveaux de détails définis sur les axes en fonction de leur diamètre, sont associés 5 niveaux de probabilité de présence de feuilles, représentant respectivement 5%, 10%, 15%, 25% et 45% du nombre total de feuilles. Pour une feuille donnée, en sortie de simulation, la définition de ce niveau de détail est simplement attribuée par le tirage d'un nombre aléatoire. Ainsi, au niveau de détail "1", un arbre "AMAP" présente donc généralement le tronc, la base de branches maîtresses et 5% des feuilles distribuées quasi équitablement dans la couronne. Ce principe est efficace, peu coûteux, d'autant qu'une écriture judicieuse des constituants -du niveau de détail le plus faible au plus élevé, facilite grandement son exploitation: plus on veut de détails, plus on lit d'éléments dans le résultat de la simulation.

Mais avec le développement de la simulation du fonctionnement, la taille des organes devient très variable, reflétant la compétition interne au sein de la plante et les conditions environnementales

avec leurs dynamiques. Seules les feuilles apparues au même moment, au même stade physiologique, sont identiques. De manière générale, il fallait reconsidérer le problème.

Nous avons développé les représentations multi-échelles du feuillage en coopération avec l'équipe de Xiaopeng Zhang au LIAMA et plus particulièrement dans le cadre de la thèse de QingQiong Deng (de 2004 à 2008). A partir de 2003, sur la base de l'idée de Remolar [LOD_Remolar02], [LOD_Remolar003], sont testé dans un premier temps les approches de fusion de feuilles sur des arbres générés par la gamme AMAP [LOD_Zhang003]. Dans cette approche, toute feuille est supposée être un quadrilatère. L'idée est de fusionner les quadrilatères entres eux, deux à deux, itérativement, selon des critères géométriques. L'union est opérée de manière simple construite à partir des points extrêmes présents dans les feuilles candidates (figure LoD_6).

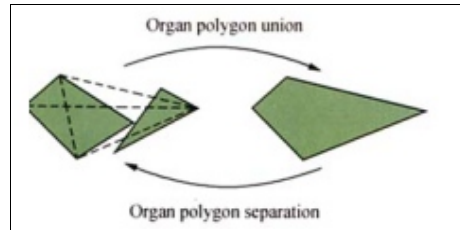


Figure LoD_6. Union de 2 feuilles (issu de [LOD_Zhang006a])

Le choix des candidats est dicté par un coût de fusion impliquant les variations d'aires, de normales, de positions. C'est le point délicat de l'approche. Une application simple conduit à charger en mémoire l'ensemble des feuilles (ce qui peut être un problème en soi) et faire une comparaison deux à deux, donc en coût $O(n^2)$. Des contributions successives nous on conduit à exploiter la connaissance de la structure, en particulier les groupes de feuilles d'une même unité de croissance, voir figure LoD_7 [LOD_Zhang006a].

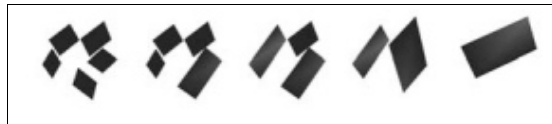


Figure LoD_7. Phases d'union de feuilles dans une unité de croissance (issu de [LOD_Zhang006a])

L'efficacité se trouve améliorée avec l'implémentation d'une approche par cluster, minimisant le nombre de candidats à l'opération d'union [LOD_Deng007a], [LOD_Deng008]. Les algorithmes sont étendus à des feuilles non quadrilatères. Nous montrons également que l'on a intérêt à développer des algorithmiques plus spécifiques lorsque l'on traite des conifères. La représentation des aiguilles, constituée par des cylindres définis par 8 faces est progressivement dégradée en 6 puis 3 faces pour être remplacée par des lignes semi-transparentes, dont on ne représentera progressivement qu'une instance parmi toutes [LOD_Deng007c].

Les opérations successives d'union sont poursuivies jusqu'à l'obtention d'un unique quadrilatère pour l'ensemble des feuilles. Cette décomposition constitue un arbre dont chaque nœud correspond à une opération de fusion, et donc à une nouvelle représentation. Sur chaque nœud on stocke également le coût de l'union. Lors de la visualisation, on parcourt l'arbre dont les nœuds ont un coût compris entre deux valeurs seuils correspondant à une erreur de projection (contrôlée par l'utilisateur et les paramètres de projection).

Il est classiquement d'usage de définir le niveau de détail (c'est à dire l'erreur acceptable pour une union) pour l'ensemble de la couronne, en fonction de la distance entre le centre de la couronne et l'observateur. Dans nos implémentations nous allons différencier ces niveaux au sein des couronnes grâce à l'usage d'informations déjà existantes: la densité des feuilles (car on connaît le nombre de celles ci par cluster) [LOD_Deng010]. En effet plus les feuilles sont denses, plus les unions seront nombreuses et pertinentes. Cette approche est favorable aux niveaux de détails intermédiaires, comme illustré ci dessous (figure LoD_8).

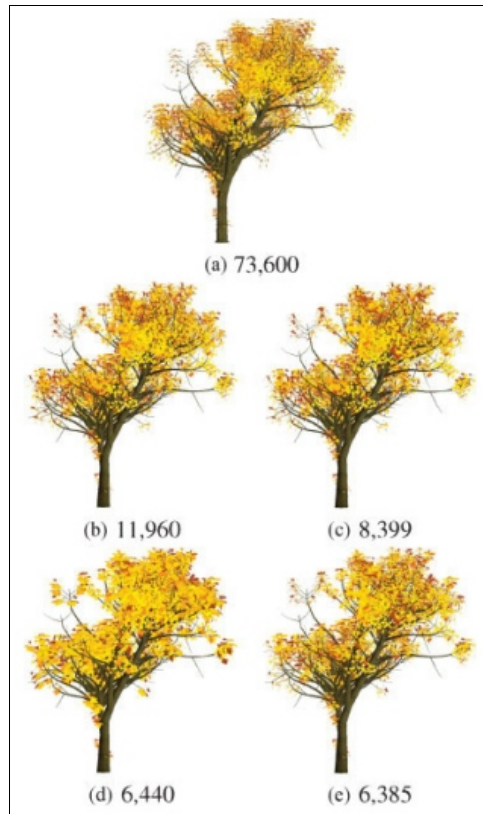


Figure LoD_8. Usage de la densité pour l'union des feuilles (a) modèle original, (b) et (d) sont deux représentations multi-échelles . (c) et (e) sont les représentations multi-échelles correspondant respectivement à (b) et (d) en employant le critère de densité (issu de [LOD_Deng010])

De même, nous différencions les niveaux de détail au sein de la couronne en fonction de la distance et de l'orientation des structures portant des feuilles [LOD_Deng007b], partant du principe que les axes "arrières" présenteront une occlusion importante (Figure LoD_9).

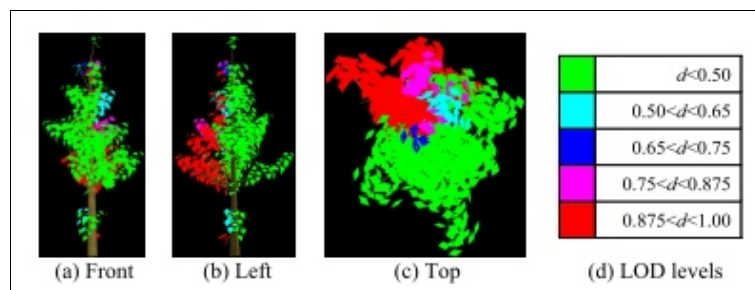


Figure LoD_9. Variation des niveaux de détail en fonction de l'orientation des axes et de leurs distance à l'observateur (issu de [LOD_Deng007b])

Les approches LoD développées sur les feuilles s'étendent aux organes fleurs et fruits.

Vers la représentation multi-échelle temps réel de forêts.

La visualisation interactive de forêts (autorisant des vues de près) utilise conjointement les approches de niveau de détail présentées ci dessus. Le synoptique général comprend donc une phase de pré-traitement créant les structures multi-échelles (les squelettes et l'arbre des unions foliaires). En phase de rendu de scènes, la géométrie des végétaux visibles (non clippés) est générée à partir du squelette pour les branches, à partir de l'arbre des unions pour les feuilles. En fait cet arbre est linéarisé et transmis à la GPU lors du chargement de la scène. Lors du rendu, seul deux index sont envoyés à la GPU, qui restitue les quadrilatères feuilles dont les nœuds sont compris entre les 2 index (figure LoD_10)

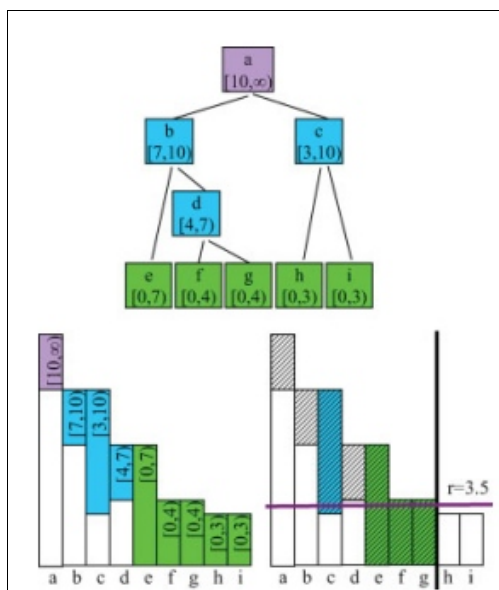


Figure LoD_10. Conversion de l'arbre des unions de feuilles en table envoyée vers la GPU. Le couple (a,b) attribué à chaque nœud est le niveau de détail min et max. A gauche la GPU est affichée le feuillage entre les niveaux 3 et 5 (nœuds c, e,f, et g) (issu de [LOD_Deng010])

Les performances de l'approches sont intéressantes; les figures LoD_11 et Lod_12 illustrent des performances de compression et d'affichage sur un arbre seul et une scène de complexité moyenne.

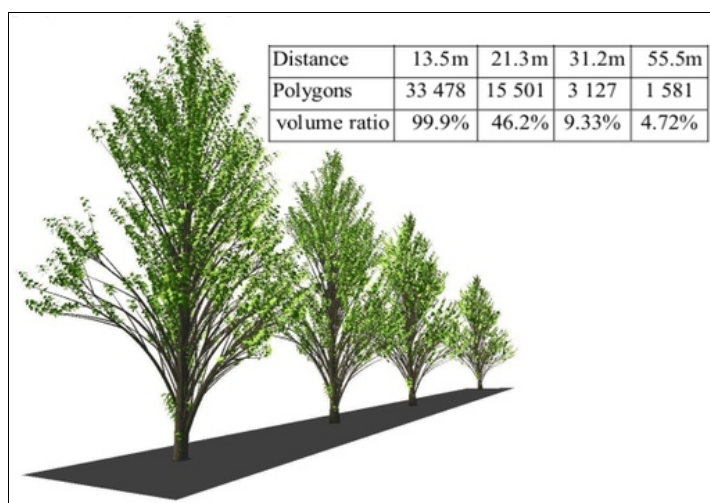


Figure LoD_11. Performances en compression (issu de [LOD_Deng009])

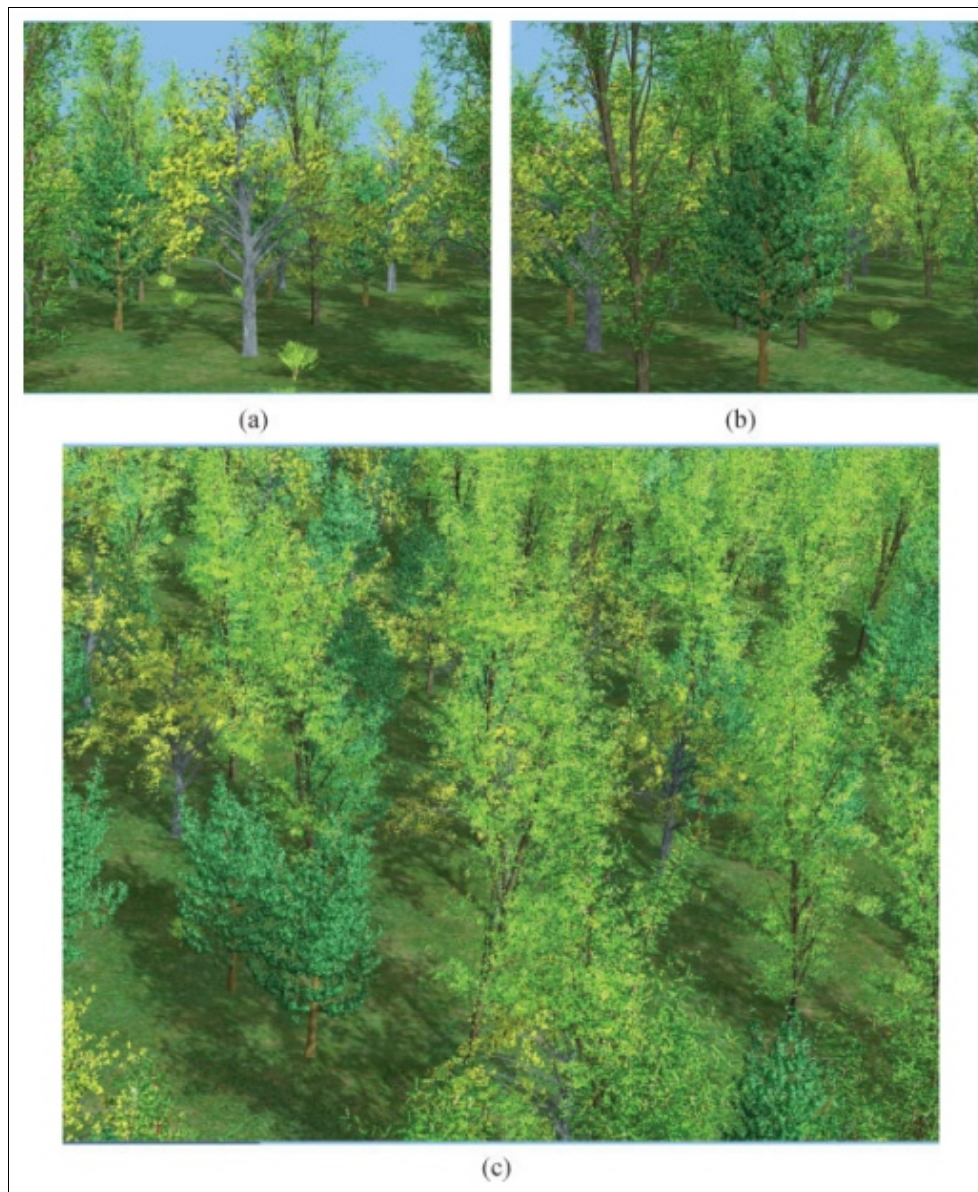


	Figure 20(a)	Figure 20(b)	Figure 20(c)
Total number of trees	16 888	16 888	16 888
Number of trees in view	279	605	311
Foliage compression ratio	0.39%	0.63%	0.50%
Number of frames per second	62.50	32.26	66.67

Figure LoD_12. Performances sur une scène de forêt (issu de [LOD_Deng010])

L'approche multirésolution ne se limite pas à la structure géométrique mais s'applique aussi à la description fine des organes et à ses attributs de rendu.

Avec HongJun LI, doctorant au LIAMA également, nous avons intégré une suite de tels modèles dans la chaîne de visualisation [LOD_Li010].

Le modèle géométrique de feuilles permet une représentation souple et continue allant d'une modélisation exhaustive tridimensionnelle de la géométrie texturée (acquise à partir de photographie de réelles feuilles), à une simple représentation par un quadrilatère plan de couleur

uniforme (figure LoD_13). Cette approche complète le modèle multi-échelle de feuillage en autorisant des vues très rapprochées réalistes.

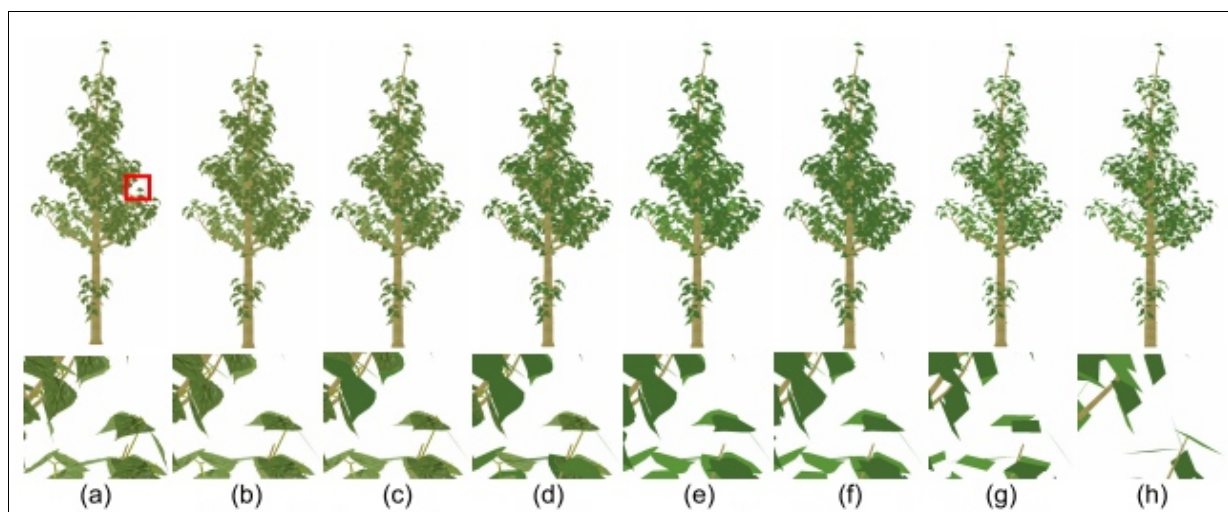


Figure LoD_13. Modèles multi-échelles de feuilles et feuillage (issu de [LOD_Li010]). Évolutions: disparition de texture (a) à (c), simplification géométrique (a) à (f), unions (f) à (h).

Les limites des approches polygonales multi-échelles

A l'issue de ces travaux, la génération de modèles multi-échelles à partir de plantes simulées ou reconstruites semble avoir atteint un stade de maturité, dont les évolutions seront sans doute faibles et basées sur un usage accru de la GPU, notamment pour la génération du maillage des branches à partir du squelette. L'approche multi-échelle polygonale reste cependant lourde. Grâce au clustering et à l'usage d'informations topologiques, les temps dédiés au post traitement n'ont cessé de diminuer pour atteindre la seconde pour les arbres illustrés ici, mais le volume consacré aux représentations multi-échelles est conséquent. Et les performances en visualisation sont effectives si l'on peut disposer en mémoire (et mémoire GPU pour le feuillage) de ces modèles géométriques. Donc si les plantes à représenter sont fortement instanciées dans les scènes. Ces approches géométriques assurent une géométrie réaliste pour des vues « pedestres ». Dans un contexte de vue éloignée (aérienne par exemple), en l'absence de vue de près, une approche basée sur des imposteurs s'impose. Mais sans pour autant pouvoir répondre au volume d'information en présence de nombreux individus distincts.

Les représentations multi-échelles basées image

Imposteurs

A l'UMR Amap du Cirad, dès les premières conceptions de scènes, il m'était devenu incontournable d'utiliser des techniques simples représentant des images de végétaux plutôt que leur géométrie. On parlait alors de "formes pixels" [MP_Jaeger987] (Figure LoD_14)

Les approches basées images, ou encore appelés imposteurs, sont devenues rapidement populaires avec le développement des fonctionnalités de gestion des textures dans les GPU et l'augmentation

des mémoires vidéo.

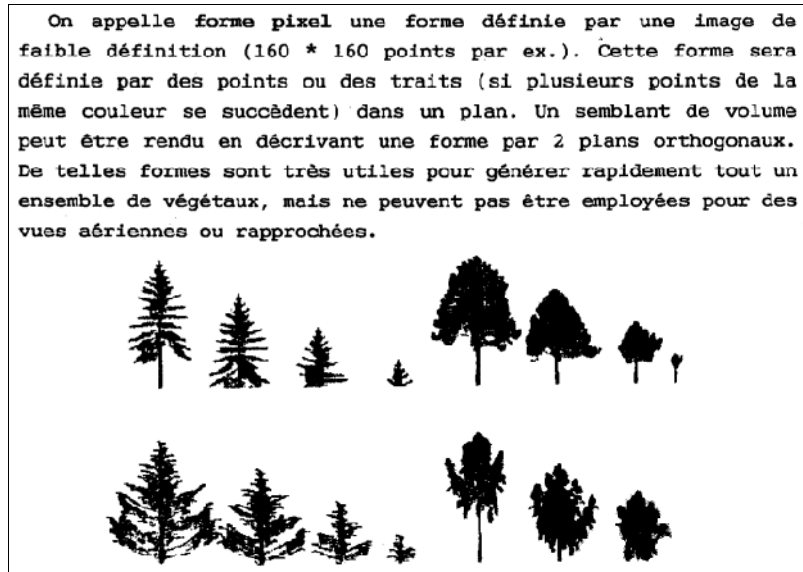


Figure LoD_14. Les formes "pixels" (issu de [MP_Jaeger987])

Après l'usage de simples images texturés avec un canal de transparence [RRB_Honjo001], encore explorées à ce jour pour la visualisation Web [RRB_Honjo010], se sont rapidement développés des techniques d'usage de plusieurs textures:

- avec des coupes de vues orthographiques [RRB_Jakuli000] ou obliques [IV_Decaudo04], [IV_Decaudo09]
- avec la gestion des profondeurs [RRB_Max996] ,
- avec des vues pré-calculées et des clusters spécifiques [RRB_Decore003] [RRB_Fuhrma005] [RRB_Fernan007b] [RRB_Fernan007a] afin d'améliorer l'illumination, incluant souvent le développement d'approches hiérarchiques adaptées à la représentation de scènes naturelles [RRB_Max999], [RRB_Deussen002], [RRB_Behren005], [RRB_Candus005]

Aujourd'hui les représentations les plus spectaculaires présentes dans le monde du multimédia et en particulier dans celui du jeu (Crysis by Crytek) ou dans le monde professionnel (LANDSIM, NatFx) emploient des modèles hybrides, qui emploient des éléments structuraux polygonaux pour le tronc et quelques axes principaux et utilisent des images pour les couronnes ou les complexes feuillés. Les techniques de rendu utilisent massivement les capacités des cartes graphiques avec, entre autres, des approches de rendu multi passes [CG_Kharla008], [CG_Sousa008].

Dans notre contexte, toujours en collaboration avec l'équipe du Professeur Xiaopeng Zhang au LIAMA, il y a simplement eu adaptation de quelques unes de ces techniques au végétal, et plus particulièrement à la plante seule. Une approche simple et efficace inspirée de [RRB_Decore003] consiste à créer un atlas avec des vues orthographique de l'arbre en entier ou de sa couronne dans le cas hybride [RRB_Bao010] comme illustré en figure LoD_15.

Une autre méthode explorée exploite les acquis des modèles multi-échelles des feuillages: on attache des textures de feuillage et branches à des plans contenant les parallélogrammes définissant le feuillage pour des niveaux de détail très faible [LOD_Deng007a].



Figure LoD_15. Atlas de textures couronne et arbre entier (issu de [RRB_Bao010])

Cette approche hiérarchique s'est révélée efficace pour la reconstruction 3D de forêt à partir d'inventaire réels tel qu'illustré en figure LoD_16 (l'ombre au sol est également un imposteur).



Figure LoD_16. Vue d'une forêt par nuage hiérarchique d'imposteurs (issu de [LOD_Deng007a])

Venons en à une approche originale, dont l'exploitation en visualisation n'était pas attendue: il s'agit du plongement du squelette des plantes dans un espace voxel.

Représentations issues de plongements dans l'espace voxel

Dans cette approche orientée image, on plonge les plantes dans un espace voxel. Cette opération traditionnellement laborieuse est grandement simplifiée ici, car on ne plonge que le squelette, c'est à dire l'axe principal de l'organe. Les grandeurs géométriques orthogonales à la direction principale (diamètre pour les axes) sont codés comme des niveaux de gris. Il n'y a pas de gestion particulière

de la compétition interne à un voxel, lors du tracé on garde l'axe dont le diamètre (le niveau de gris) est le plus élevé. Le plongement est donc un tracé de segment de droite en niveau de gris et s'implémente par un algorithme du type Bresenham 3D [CG_Bresen965]. A l'issue du plongement on dispose de deux espaces voxels; l'un de labels l'autre de diamètres (ou biomasse). L'intérêt de cette approche est d'accéder à une représentation indépendante de la simulation et d'évaluer la biomasse simplement, par simple sommation des contenus des voxels [IV_Jaeger003].



Figure LoD_17. Plongement d'une scène dans un espace voxel (issu de [IV_Jaeger003])

Ce plongement trouve aussi un intérêt pour la visualisation. Le plongement peut facilement être visualisé malgré la mauvaise représentation de l'épaisseur des axes (Figure LoD_17). Celle ci peut être restituée par des transformations euclidiennes sur l'espace discret (EDT), en simulant une diffusion des niveaux de gris [TIS_Borge986]. Nous avons implémenté cette dernière transformation par une approche récursive séparée, à temps constant en parcourant l'espace voxel selon plusieurs axes.

Après application d'un tel opérateur de dilatation, l'espace voxel peut être visualisé selon des techniques classiques d'imagerie volumique. L'usage de fonctions de transfert [IV_Pfiste001] utilisant la transparence permet d'obtenir des représentation réalistes en particulier sur les branches de faible diamètre. Dans un espace discret isotrope, l'implémentation de modèles d'illumination avec de l'ombrage et des réflexions est possible par un lancer de rayon discret, en employant, une fois de plus un tracé de rayon dérivé d'un algorithme de Bresenham (Figure Lod_18)



Figure LoD_18. Vue réaliste d'une scène voxel (collection de l'auteur , 2005)

Contrairement à l'ensemble des autres approches de la littérature, l'approche n'est pas contrainte par la diversité du contenu de la scène mais uniquement par la résolution de l'espace voxel (limité à 4 Giga voxels pour des questions d'adressage). On peut ainsi créer des espaces voxels contenant plusieurs centaines de plantes toutes distinctes les unes des autres (Figure Lod_19) et obtenir des visualisations dans des temps de l'ordre de la minute.



Figure LoD_19. Une scène voxel constituée de plus de 500 plantes toutes différentes l'une de l'autre. Image annonce de PMA06 <http://www.pma.cirad.fr/PMA06> (collection de l'auteur , 2005)

L'évolution des approches LoD pour la visualisation des plantes et des paysages

Que ce soit pour la structure ou pour les organes (feuillés ou floraux), on peut aujourd'hui considérer que les techniques de fusion/simplification des structures, de la définition de leurs géométries et de leur représentations graphiques ont atteint un degré de maturité stable, performant, mais sans grande marge de progrès, hormis ceux techniques apportés par l'usage intensif des possibilités de la GPU et de son environnement. Il en est de même pour les représentations à base d'imposteurs dont les principes d'application sont clairement établis en construisant des clusters hiérarchiques sur la base des acquis en compression du feuillage ou sur la base des sous-structures définies lors de la simulation.

La simplification et la compression des représentations géométriques exhaustives (fondées sur la liste des constituants de la plante telle que la ligne élastique) a donc dévoilé son potentiel et atteint ses limites. La définition de nouveaux modèles performants de représentation et visualisation de végétaux simulés (voire reconstruits) doit donc s'appuyer sur des représentations de plus haut niveau, définies en amont au niveau de la simulation de la structure, voire du fonctionnement. La difficulté va résider dans la portabilité des approches suggérées. En effet, partir de description de plus haut niveau revient à dire que la géométrie sera reconstruite lors du rendu; et cette reconstruction ne devrait pas se transformer en simulation, mais rester dans un contexte ouvert à de multiples approches de génération de végétaux, avec une complexité faible, susceptible d'être exécuté sur la carte graphique.

CHAPITRE III

Les paysages dits fonctionnels

Nous avons introduit plus haut le terme de paysages fonctionnels en référence à l'imagerie médicale fonctionnelle. Nous détaillons ici un peu plus ce point mais ne rentrerons pas dans les détails de modélisation et simulation des modèles bio-physiques qui s'y rattachent.

De la plante à la culture et au paysage avec un même modèle de plante ?

En agronomie tout comme en foresterie, la production -au sens économique du terme- n'est pratiquement jamais développée à l'échelle de la plante seule. Selon les systèmes de productions, les cultures sont organisées en unités (serres), champs, plots et peuvent s'inscrire sur un espace géographique relativement vaste. Plus généralement, ce sont à la fois les critères de ressources et d'itinéraires cultureux qui en fixent les contours.

Les modèles de production exploités depuis plusieurs décennies ont montré leurs fiabilités sur un grand nombre de cultures dans un environnement stable et sur des cultures homogènes mais aussi leur faiblesses à prendre en compte la dynamique des systèmes et toute évolution dans les ressources ou itinéraires de cultures. D'où une tendance à explorer des grains plus fins, orientés sur l'individu plante, en interaction avec son environnement. Le modèle Greenlab, dans son expression actuelle [MP_deReffye008a], bien qu'historiquement défini à l'échelle de la plante, peut être exploité efficacement à ces échelles dans un contexte mono-spécifique (une seule culture). Son expression mathématique tant au niveau de la quantification de la production que de la quantification de l'expression de la structure permet (lorsque l'interaction entre les deux principes reste simple) de multiples applications grâce au contrôle.

Un système complexe

A des échelles plus importantes, ou tout simplement lorsque l'on s'intéresse à des zones de production non homogènes (tant en terme d'espèces, que d'itinéraire de culture ou en présence d'importantes variations de ressources), aujourd'hui, seuls des modèles de simulations permettent d'explorer la dynamique complexe de ces systèmes. Dans ces approches, ce qui est malheureusement le cas d'un nombre croissant de situations pratiques, on se doit d'étudier le partage des ressources (de l'eau spécifiquement). A la compétition des organes en interne s'ajoute donc la compétition "externe" liée aux ressources. Or le végétal impacte cette ressource. Sa disponibilité tant temporelle que spatiale se doit donc d'être prise en compte dans la dynamique du système de culture étudié. Ce qui veut dire les simulations doivent non seulement intégrer des instances différentes d'un même modèle de culture mais aussi intégrer des modèles d'évolution des ressources et donc intégrer tout modèle qui contribue à modifier significativement la disponibilité de la ressource en espace et en temps.

Dès aujourd'hui, bon nombre de questions mobilisant la production végétale et son impact environnemental pour ne pas dire plus, impose d'inscrire les travaux de modélisation dans un contexte méthodologique permettant l'interaction de nombreux modèles bio-physiques (voire sociaux) à travers des ressources partagées conservatrices ou non. Il est donc crucial, à relativement court terme, de pouvoir exprimer un modèle tel que GreenLab, sous une forme qui permettent l'interface avec d'autres modèles biophysiques en géologie, hydrologie, climatologie, d'écologie, dans le but de modéliser les processus et les dynamiques de ruissellement et de prélèvement de la ressource en eau, de la prise en compte d'évolutions climatiques et plus généralement d'impacts locaux.

Il est donc fondé de développer de s'intéresser aux échelles couvrant la culture/plot jusqu'au paysage, échelles dans lesquelles l'approche formelle Greenlab est spatialisée; c'est à dire appliquée

localement (dans des zones considérées comme homogènes ou, à défaut, à l'échelle de la plante seule) interagissant avec les ressources – en eau.

Cet axe d'étude, que l'on peut qualifier de modélisation et simulation bio-physique de systèmes complexes, pose aussi des problèmes intéressants de représentation visuelles de données, qu'elles soient simplement des paramètres de simulation, des niveaux de ressource ou de production, de dynamiques, d'indicateurs. Mais ce rôle de représentation va bien au-delà de la simple exploration.

En effet, la communauté graphique joue un rôle moteur dans les approches de modélisation et simulation de tels environnements virtuels

Les approches originales de la communauté graphique

Dans la dernière décennie, avec les technologies des cartes graphiques (GPU) et les approches de rendu orientée images [CG_Meyer001], la représentation de scènes naturelles plus ou moins homogènes en temps réel est devenue commune. Avec des modèles d'illumination ad-hoc [CG_Coden004] et l'usage d'imposteurs l'atmosphère simulé des scènes est convaincant. Ce réalisme est souvent contextuel. Il ne privilégie pas la dynamique propre au composant végétal. Lorsque l'on s'écarte cependant de ces contraintes que l'on qualifie volontiers de "temps réel", force est de constater que la définition procédurale de biotopes, -même dans des contextes d'application d'imagerie stricte a toujours mobilisé les équipes de référence en synthèse d'image.

La définition de "biotopes"

A l'échelle de vaste paysage, pour des vues en survol, une des méthodes répandues consiste à définir des biotopes auxquels on attribue des textures matérialisant le couvert végétal. Les travaux pionniers de Hammes [MSL_Hammes001] représentent ainsi des scènes naturelles par un jeu hiérarchique de textures. Dans de nombreuses applications, cette attribution est définie selon une carte à faible résolution spatiale [MSL_Griffon009] ou automatisé en fonction de données relatives au terrain [ST_WCS010][ST_Terragen010], [ST_Forester010]. On peut également définir une description un peu plus exhaustive en terme de composition, fondée sur des statistiques de distributions d'espèces avec leurs couvertures respectives [ST_Bionatics010] [MSL_Coligny007]. La dynamique des cartes (rotation des cultures) peut elle même être modélisée formellement, via des grammaires stochastiques [MSL_Gaucherel006]. Ces approches sont parfois combinées.

Représenter et simuler la compétition

La constitution de biotopes plausibles, issus d'un processus d'évolution / de compétition sur une surface définie fut abordée très tôt par Ned Greene avec une approche voxels, proche d'un automate cellulaire 3D [MSL_Green989]. Cette modélisation de la compétition -uniquement destinée à représenter des images réalistes- devient une préoccupation d'équipes renommées, alimentées en particulier par les contributions de P. Prusinkiewicz et O. Deussen [MSL_Deussen998]. L'obtention de paysages de synthèse comme causalité de phénomènes bio-physiques couvre en fait deux processus bien distincts: ceux contribuant à définir une échelle macroscopique – visant à modéliser la surface "brute" qui simulent des érosions (hydrologiques/éolienne) [MSL_Neidhold005], et ceux cherchant à construire le couvert de cette surface, mais à une échelle plus fine, visant à définir le couvert végétal à l'échelle locale, jusqu'à la compétition entre individus [MSL_Alsweis005].

Simulation et visualisation de la compétition avec GreenLab

Dans le modèle GreenLab, la compétition peut être simulée selon deux approches: en jouant sur la surface de développement (donc sur la densité) et/ou en jouant sur le facteur environnemental (donc cette fois plutôt sur les ressources en eau). Si l'on suppose les ressources non limitantes, on peut définir une surface de projection individuelle en fonction de la distance au voisinage.

Nous avons réalisé un premier essai qui s'appuie sur une chaîne d'outils : un éditeur de scène [ST_Simeo2010] issu de l'outil de sylviculture CAPSIS [MSL_Coligny007], un mailleur de Voronoi incrémental (issu de la chaîne d'imagerie médicale), le simulateur Digiplante implémentant le modèle GreenLab et enfin la visualisation des paysages donc les principes ont été décrits plus haut. Le mailleur de Voronoi fournit la surface de développement associée à chaque individu de la scène. Cette spatialisation définit un pavage simple; c'est une version simpliste de constructions de couvertures spatiales telles que définies par D. Hill [MSL_Hill994]. De la distribution de ces valeurs de surface sont définies des classes de compétitions (de densité en fait), et pour chaque classe, un représentant est calculé. [MSL_Cournede010] (Figure Pf_1)

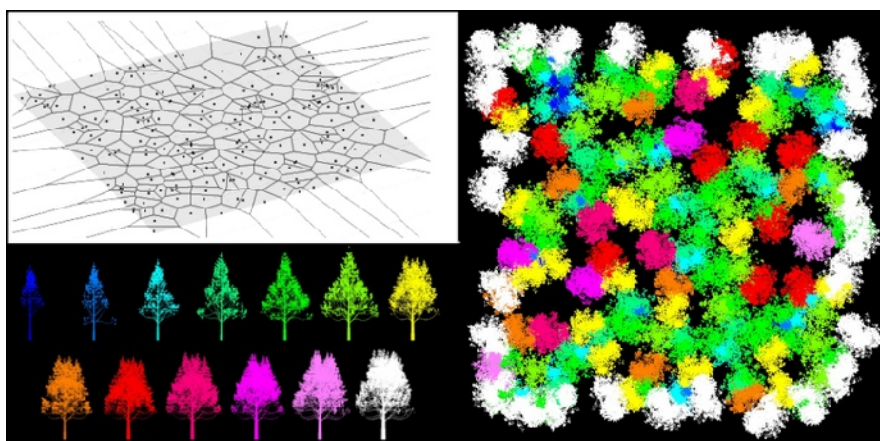


Figure Pf_1. Classes de compétition sur une forêt équiennne. (issu de [MSL_Cournede010])



Figure Pf_2. Visualisation de scène avec compétition (issu de [MSL_Cournede010])

Les sorties sont ensuite envoyés au visualiseur pour une représentation réaliste (Figure Pf_2). Dans ce travail, le développement des plantes en compétition n'influe pas le milieu, en particulier sur la disponibilité des ressources. Il n'y a pas de rétro-action entre les plantes et le sol. L'imagerie s'appuie sur une représentation classique des structures, en employant les modèles géométriques multi-échelles décrits plus haut. La chaîne de simulation/visualisation est lourde, séquentielle, sans interaction avec l'utilisateur. Elle montre un intérêt pratique, car opérationnelle, basée sur des techniques éprouvées, et reste la seule approche susceptible d'être validée par l'expérimentation sur le terrain, en attendant la maturité des approches de simulations fonctionnelles des paysages.

Le développement d'outils de simulation et visualisation de paysages fonctionnels

Nous avons vu qu'en plongeant les squelettes des plantes calculées dans l'espace voxel, nous quittons les représentations de la géométrie des constituants individualisés (plantes, sol, etc, ...) pour une représentation de l'espace lui même, dont les composants, les voxels, contiennent une grandeur fonctionnelle proportionnelle à une teneur en eau [IV_Jaeger003]. L'idée est de faire circuler cette eau, selon des lois physiques, à l'image des travaux pionniers de Ned Greene [MSL_Green989]. Cette démarche de développer des simulations de croissance de plante dans l'espace voxel est exploratoire, d'autant plus qu'elle nécessite d'ouvrir notre culture à des domaines que nous ne maîtrisons pas, telle que l'hydrologie. Dans ces travaux, nous faisons des simplifications drastiques, en particulier sur la spatialisation du modèle de plante que l'on considère comme limitée à un unique voxel, tant sur la partie structure que sur la ressource en eau avec laquelle la plante interagit.

Une première approche dans un espace voxel

Le cadre du LIAMA, de par la présence de chercheurs impliqués dans des travaux liés aux simulations d'inondations est un cadre propice pour initier ces travaux. Avec Aurélie Lesluye en stage de Master « Systèmes Complexes » de l'Université de Chalmers (Suède), nous construisons au LIAMA un premier prototype, simple, basé sur un découpage homogène en voxels de l'espace.

La simulation mime le cycle de l'eau, sur un pas journalier, selon les processus séquentiels suivants:

- les précipitations
- l'absorption (première couche de voxels dans le sol)
- le ruissellement (traité comme une diffusion)
- la diffusion dans le sol
- l'absorption par les plantes
- l'évaporation (l'évapo-transpiration pour les plantes).

Chaque processus est modélisé de manière simpliste: le modèle de plante est un axe feuillé à expansion immédiate, sur un cycle journalier. Le processus de ruissellement est traité comme une diffusion. L'utilisateur peut, à chaque étape, influencer les processus en modifiant des paramètres caractéristiques tel que le niveau de précipitation, ou encore un coefficient de diffusion dans le sol.

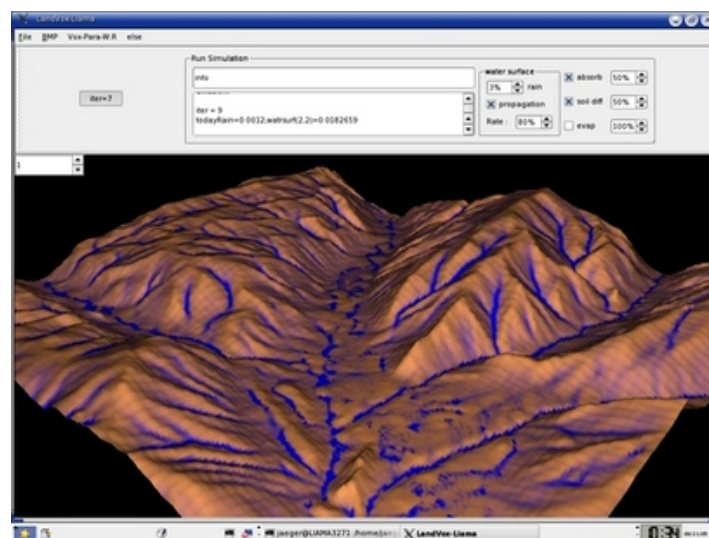


Figure Pf_3. Premier prototype de simulation fonctionnelle (issu de [MSL_leChevalier007a])

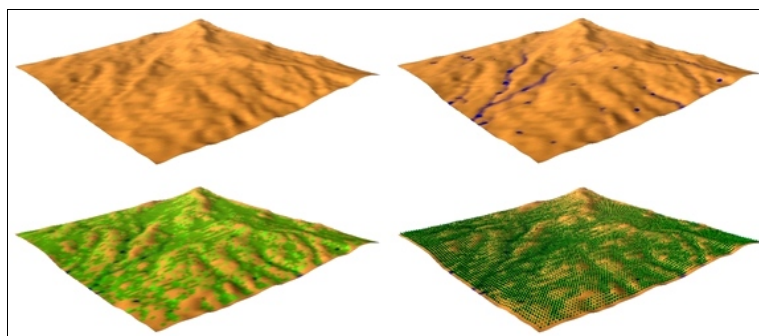
La simulation s'effectue en explorant l'espace des voxels, que l'on parcourt de manière différenciée, en déroulant les processus, tout à tour. Chaque processus met à jour la ressource en eau sur le voisinage local. L'approche est donc rustique, biaisée et physiquement non valide; elle permet cependant de représenter des effets de compétitions et d'influence de l'environnement. Développée dans l'environnement libre Qt [ST_QT010], l'application est opérationnelle sur de simples PC portables comme sur des calculateurs parallèles en dédiant un processeur à chaque processus par l'usage de canaux (threads) [MSL_leChevalier007a] (Figure Pf_3).

La visualisation de la simulation est dans un premier temps déportée, l'espace voxel est exporté et le rendu effectué avec des fonctions de transfert par les outils de rendu volumique (Figure Pf_4)



Figure Pf_4 Visualisation volumique de paysage simulé (collection de l'auteur, 2006)

Nous avons ensuite développé des outils de visualisation spécifiques, orientés images. La géométrie est définie par le modèle numérique de terrain. Une texture est calculée à partir de l'illumination (simple lambertienne) de la surface du terrain que l'on combine avec de la transparence à la présence d'eau en surface puis à celle de la biomasse. Un variante consiste à visualiser la biomasse selon une sphère avec un rayon proportionnel à la valeur de la production cumulée (Figure Pf_5). La texture est recalculée à chaque cycle, l'évolution de la simulation est donc suivie pas à pas.



*Figure Pf_5 Visualisation d'un paysage fonctionnel simulé. (issus de [MSL_leChevalier0007a]).
En haut: terrain seul; ajout de l'eau en surface. En bas, à gauche: ajout de la biomasse cumulée.
En bas à droite: cette biomasse est représentée par une sphère sur chaque voxel en surface.*

Ce premier prototype empirique met en évidence la nécessité de construire une approche de simulation composée de modèles plus réalistes, et de développer davantage le potentiel de la visualisation. Le découpage homogène de la scène en voxels pose par ailleurs de nombreux problèmes pour l'exploitation des modèles hydriques, dont le formalisme et les algorithmiques sont adaptés à des découpages de l'espace en strates de profondeurs définies à partir de la surface du sol.

Un nouveau prototype pour la simulation de paysages

A partir de cette première expérience, nous développons un deuxième prototype dans le cadre de la thèse de V. Le Chevalier (ECP). Il est construit sur une grille discrète homomorphe à la définition du terrain. Nous quittons donc l'espace voxels pour des strates de profondeur. Cet outil comprend un nouveau modèle de ruissellement, définit localement et récursivement. Dans cette approche, la vitesse de l'eau est supposée infinie, de manière à obtenir un régime stabilisé sur un pas journalier. Cependant, en profondeur, les diffusions latérales sont ignorées. Le modèle de plante est le modèle GreenLab, intégral. Son implémentation est adaptée de telle manière à ce que la prise d'eau soit journalière, avec un cycle de développement classiquement calqué sur le temps thermique. A l'initialisation, la végétation est homogène, semée sur l'ensemble du relief, avec une instance par élément de grille. L'application est ouverte aux entrées climatiques (précipitations, température). La température est localement modifiée en fonction de l'orientation et l'altitude.

Ce nouveau prototype ne comporte pas de visualisation intégrée, mais exporte, à chaque pas de simulation (le jour), des cartes rasters homogènes de valeurs calculées: biomasse journalière produite, biomasse cumulée, cycle en cours dans la culture, teneur en eau dans la première couche du sol, quantité d'eau sur le sol, température, etc... soit entre 15 et 20 cartes selon l'application.

Nous développons un outil spécifique visualisation. Il permet de représenter simultanément 4 canaux distincts, mélangeables. Chaque canal est susceptible de convertir les grandeurs scalaires en grandeurs géométriques (surface du style terrain, barre verticales, sphères de rayon variable, systèmes de particules style pluie..) ou en grandeurs colorimétriques (on associe une palette de couleur et un mode de mapping) ou en grandeur de combinaison avec un autre canal (transparence, mixage de couleurs ..) [MSL_leChevalier0007b] (Figure Pf_6)

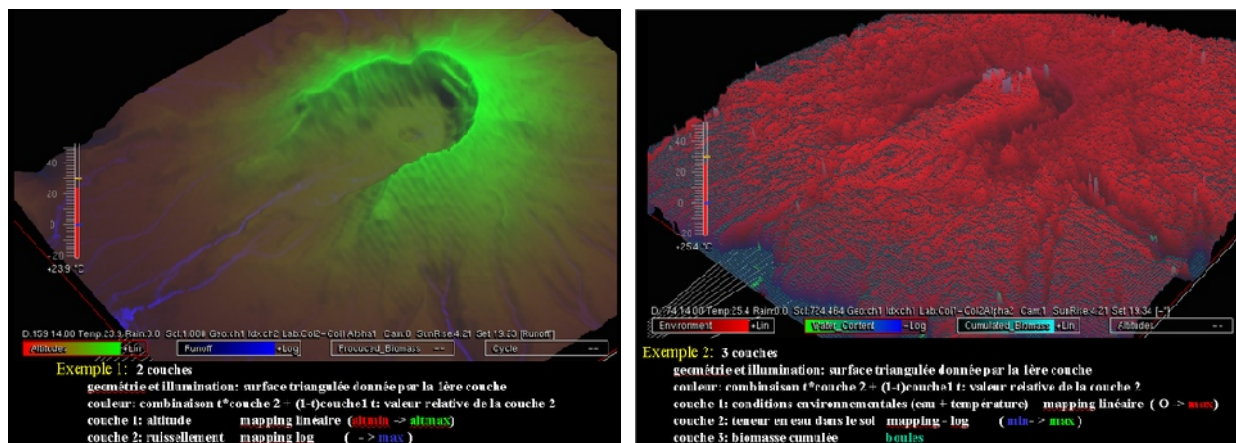


Figure Pf_6. Visualisation fonctionnelle d'un paysage (collection de l'auteur, 2008). A gauche, terrain réaliste avec le ruissellement. A droite, facteur environnemental (rouge) avec la teneur en eau dans le sol (bleu) et la biomasse cumulée (sphères bleues)

L'outil présente un tableau de bord précisant les modalités d'affichage et les conditions climatiques (Figure Pf_6). Il permet également de lancer une analyse des canaux sélectionnés en intégrant les variables sur la totalité de l'année, et, pour le canal en cours, d'évaluer la dispersion des valeurs sur l'ensemble du terrain à un jour donné (Figure Pf_7)

L'outil de visualisation interactif autorise également des représentations réalistes en combinant de manière adéquate la représentation géométrique du terrain avec les représentations fonctionnelles d'autres champs. On affecte dans un premier canal le terrain avec une palette de couleur fonction de

l'altitude. Puis on ajoute le ruissellement (cf Figure Pf_6 gauche). On ajoute enfin la biomasse cumulée que l'on représente par des sphères vertes.

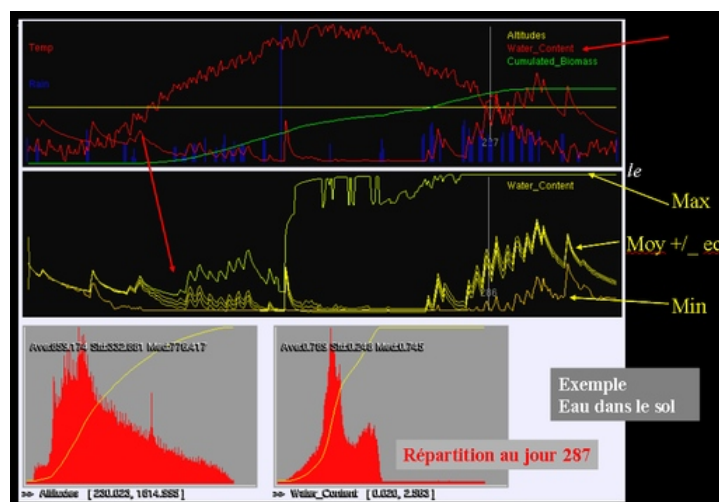


Figure Pf_7. Suivis des canaux dans un paysage fonctionnel ([MSL_leChevalier0007b]). En haut: courbes moyennes des valeurs sur l'année. Au milieu, exploration d'un canal (teneur en eau dans le sol), avec ses amplitudes moyennes sur l'année et sa dispersion sur le terrain au jour 287 (en bas).

Un héliodion est implémenté, permettant d'obtenir une direction d'éclairage cohérente et de calculer la couleur du ciel. On peut activer les fonctions de rendus en post-traitement en les couplant (empiriquement) aux canaux: évaporation pour la brume, système de particules pour les précipitations. Aux vues on ajoute enfin les fonctions d'ombrage et de profondeur de champ (voir premier chapitre).

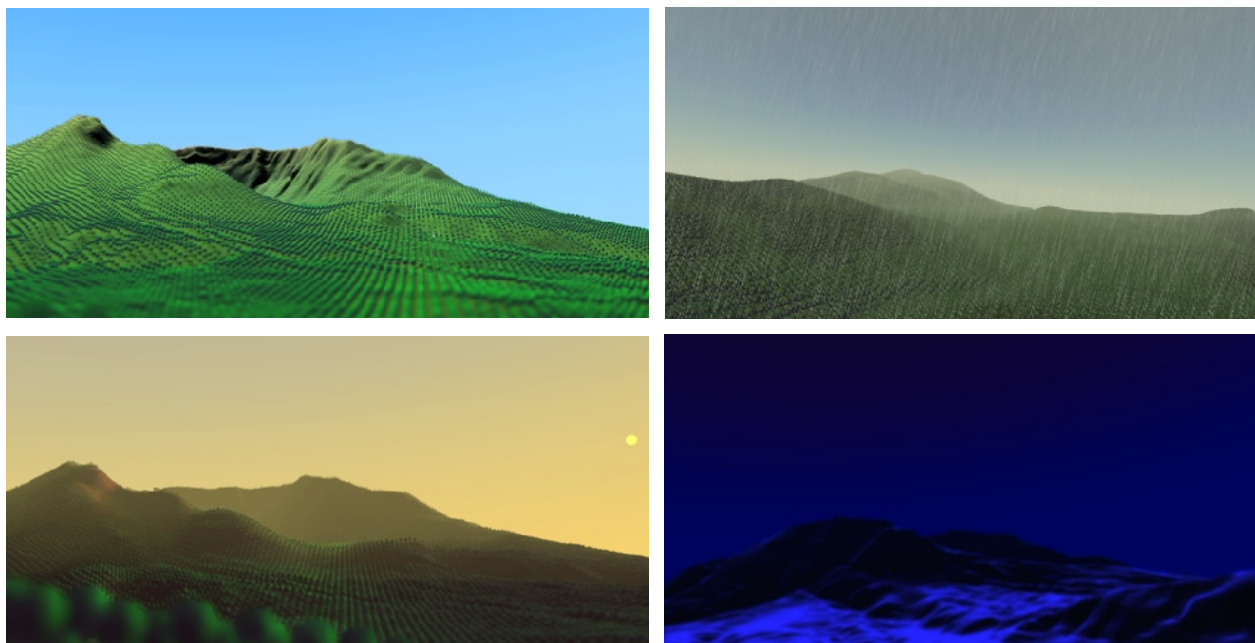


Figure Pf_8 Représentations réalistes d'un paysage fonctionnel (collection de l'auteur, 2009). De gauche à droite, de haut en bas: 16 Mai, 14:00, 15 juillet 15:00 (pluie), 20 Octobre 19:00, 28 décembre 02:00 (neige la veille).

La formalisation autour du paysage fonctionnel

Bien que prometteurs, les simulateurs développés présentent des grandes lacunes conceptuelles.

La synchronisation des modèles

La première est liée aux problèmes de synchronisation des modèles. Cela est particulièrement illustré par le modèle de plante et le modèle de ruissellement. Ces deux modèles vont affecter la teneur en eau du sol et définir des sorties issues de cette grandeur. Ils interagissent mutuellement à travers une compétition. S'il l'on veut avoir une approche modulaire constructive (c'est à dire ne pas écrire explicitement l'interaction entre les deux modèles), il faut que ces modèles puissent se synchroniser sur l'échange des données (teneur en eau) à des dates clés.

C'est une condition nécessaire mais non suffisante. Il faut que les modèles puissent évaluer leur action à chaque intervalle de temps. Ainsi, le modèle de plante, exprimé classiquement par des cycles de croissances définis par le calendrier thermique, devrait être réfléchi différemment, en permettant la croissance des organes (et donc le » pompage de l'eau«) à des dates quelconques.

Cette considération de synchronisation des modèles vis à vis des données nous ont conduit à définir un cadre méthodologique conceptuel nouveau, proche de celui de DEVS (développé par Ziegler) [MSL_Ziegler000] et à la définition d'une architecture de construction de plateforme de simulations [MSL_leChevalier0010a].

Dans notre approche, on sépare spécifiquement les processus (des modèles) des données par la définition de caches. Ces derniers contiennent une instanciation de la valeur de la donnée (ou de sa variation). Les mises à jour des données sont effectuées sous la conduite d'un modèle spécifique, le Manager, qui va définir la date de la prochaine étape de simulation en interrogeant les modèles de l'application. Les modèles vont ensuite effectuer leur simulation avec les données en cache. Le Manager actualise ensuite les données et les caches (Figure Pf_9).

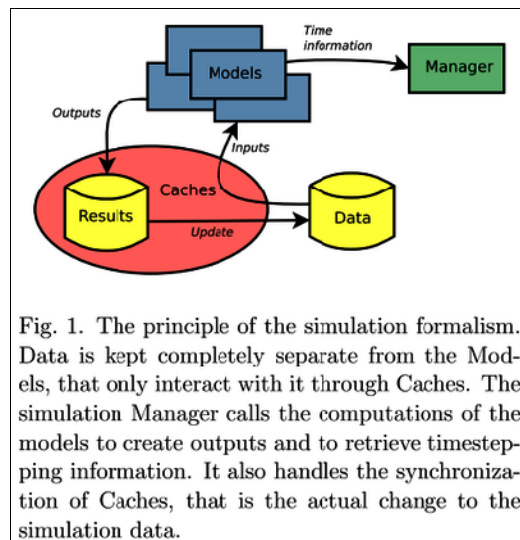


Figure Pf_9. Principe de synchronisation entre modèles et données (issu de [MSL_leChevalier2010b])

L'approche sépare de fait le calcul des évolutions des données. Le cadre générique spécifié permet

aussi une modélisation hiérarchique des modèles dans la définition des simulateurs. Elle entraîne une définition hiérarchique des caches et, in fine, des données elle-mêmes.

La modélisation de la compétition par les ressources.

La deuxième difficulté rencontrée concerne la modélisation de la compétition vis à vis de la ressource, celle en eau en ce qui nous concerne. Il nous a paru important de conceptualiser cette notion de ressource amenée à être manipulée par de nombreux modèles. Pour ce nous avons défini une classes de modèles particulier, les conteneurs [MSL_leChevalier2010a]. Ils sont définis par une capacité, une valeur de ressource et un flux de sortie (Figure Pf_10).

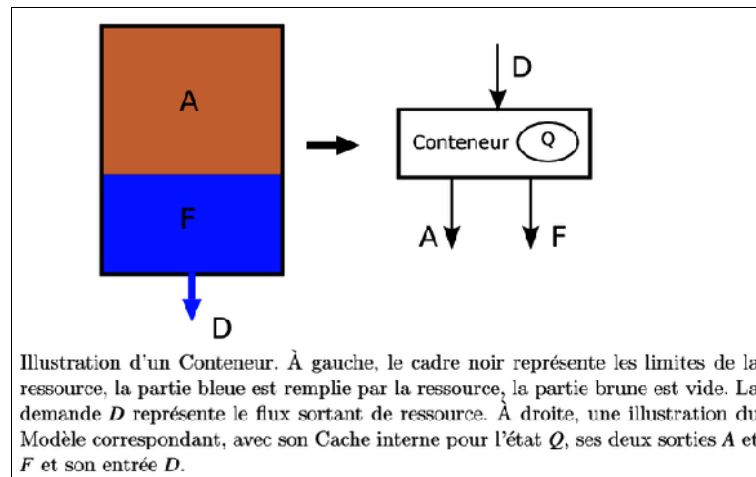


Figure Pf_10. Un conteneur pour la modélisation de la ressource (Vincent le Chevalier, 2010)

Les modèles (c'est à dire les processus) vont accéder aux informations concernant la ressource via des accesseurs – dans les modèles de plantes il est important de connaître la disponibilité en eau ce qui va définir le niveau de production. Ces derniers seront connectés aux conteneurs. En pratique, il est intéressant d'introduire des conteneurs virtuels (concentrateurs/dispatchers). Cela permet en particulier de gérer efficacement la spatialisation de la ressource et de connecter des processus sur des conteneurs partagés sans expliciter ce partage à l'échelle des processus.

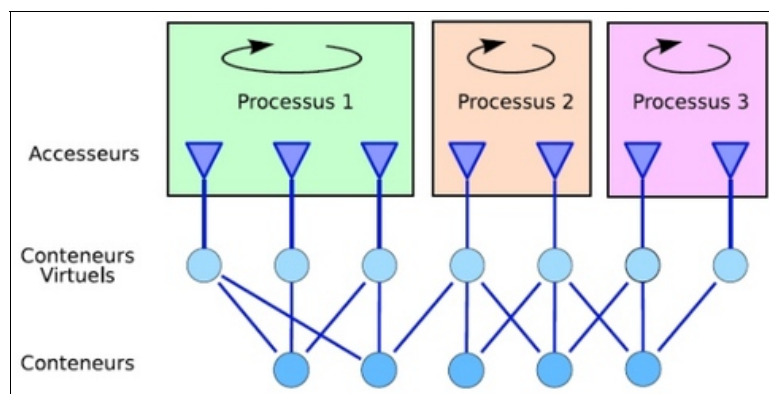


Figure Pf_11. Processus en compétitions pour des ressources (issu de [MSL_leChevalier010b])

Le formalisme développé est implémenté dans une architecture orientée objet. Elle a été illustrée sur des applications simples: pompes d'irrigation spatialisées et implémentation d'un modèle

GreenLab -simplifié- décrivant la production par une équation différentielle (au lieu d'une sommation discrète par cycle).

Les développements à venir

Les prototypes et études sur le formalisme de description de modèles pour la simulation de paysages fonctionnels montrent que ces travaux débordent largement du cadre de la modélisation de la structure et du fonctionnement des plantes, de leur représentation géométrique et de leur visualisation. Le premier enseignement que nous pouvons en tirer est que les modèles de plantes existants au démarrage de ces études ne satisfaisaient pas les contraintes de synchronisation pour permettre leur usage dans la simulation de compétitions à travers des ressources partagés et dynamiques. Le deuxième enseignement est que les paysages fonctionnels sont des systèmes complexes pour lesquels nos contributions sont (et doivent être) limités aux domaines de nos compétences: à savoir les plantes et, plus marginalement la visualisation. Avis par ailleurs partagés par nos collègues podologues, écologues, hydrologues...

Dans ce cadre, l'évolution de ces travaux est simple à exprimer: il consiste d'une part à revoir l'expression de modèles simulant la croissance, et, d'autre part, de contribuer, au sein d'une communauté élargie, à l'émergence de modèles conceptuels et d'environnements techniques pour la modélisation et la simulation d'écosystèmes, s'inscrivant dans la dynamique des travaux des pionniers et spécialistes comme P. Coquillard et D. Hill [MSL_Coquillard997] [MSL_Hill009]

Sur le premier volet, la refonte du modèle de croissance est aujourd'hui en cours selon deux modalités: l'une en développant une expression continue du modèle sous forme d'équations différentielles, menée à l'Ecole Centrale de Paris [MP_Li010] et inspirée de ce travail; l'autre sur une expression discrète calqué sur le temps calendaire, aussi fin que voulu, avec des pauses explicites entre les organogénèses successives (menée à l'UMR Amap, initiée par des problématiques pratiques issues d'études conduites sur des cultures tropicales). Ces études ne sont pas finalisés à cette heure et engendrent de nombreux questionnements, notamment en calibration et validation, du moins pour la version continue.

Sur le deuxième axe, les concertations interdisciplinaires se multiplient, principalement sous la forme d'animations scientifiques [MSL_LandMod010]. Ainsi, pour l'heure, il n'y a plus de développements en cours sur la simulation des paysages fonctionnels en tant que tel.

Il n'est pas évident qu'il faille capitaliser sur l'architecture de simulation telle que présentée ici. Une alternative intéressante, issue des interactions disciplinaires croisées, est d'évaluer l'implémentation du formalisme de la gestion de la ressource dans un environnement DEVS, celui de VLE en particulier [MSL_Quesnel993], avec la collaboration de l'INRA-BIA dans le cadre du projet RECORD. Cette stratégie est d'autant plus fondée qu'un certain nombre de collègues impliqués dans des modélisations en hydrologie (l'INRA LISAH) ou dans les sciences sociales (CIRAD-GREEN) sont dans le même schéma de réflexion. Une autre alternative intéressante serait de s'appuyer sur l'environnement DEVS SimStudio [MSL_Touraille010], développé au LIMOS.

Enfin, les fonctionnalités de visualisation développées peuvent continuer à être valorisées à court terme et mobiliser d'importants moyens d'ingénierie. Pour l'heure le visualiseur multi-canaux est en phase de tests avec les partenaires cités ci dessus pour représenter d'autres dynamiques calculées au sein de paysages: en hydrologie avec l'INRA-LISAH. Des collaborations avec des spécialistes de la modélisation des services écologiques se définissent également autour de l'usage de ces outils, avec l'Université de Dresde en particulier.

CHAPITRE IV

Vers de nouvelles méthodologies

pour représenter et visualiser

les plantes

Dans l'organisation des thématiques de recherches énoncées dans notre introduction, nous avons défini trois axes thématiques de recherches se positionnant entre les différents niveaux de représentations, selon un gradient allant du structurel (la description géométrique de l'organe) au fonctionnel (la représentation de la production dans le paysage) :

- la définition de méthodes et approches multi-échelles décrivant la structure de la plante
- la définition de géométries attachés au fonctionnement de la plante et leurs représentations
- la simulation de la dynamique des paysages et sa visualisation

Dans le cadre de ces trois thématiques, nous avons identifié deux axes de recherche proches de la modélisation géométrique (la reconstruction 3D depuis un graphe de simulation et la représentation fonctionnelle) et deux autres plus techniques, proches de la restitution visuelle (les développements GPU et les développements WEB).

A ces quatre directions de recherche sont et seront attachés des développements d'ingénierie et de valorisation conduits dans le cadre de projets en partenariat. C'est typiquement le cas de la visualisation des paysages fonctionnels – que l'on peut exploiter dans de nombreux contextes de visualisation de champs bidimensionnels de données rasters dynamiques.

Sur la thématique des représentations multi-échelles

Nous avons souligné à plusieurs reprises que la définition de représentations multi-échelles à partir de squelettes de plantes reconstruites ou simulées est arrivée à maturité et entre donc dans des phases d'ingénierie et de valorisation. Dans ces approches de décimation/compression, des gains sont encore possible par l'utilisation de la GPU, pour la génération des primitives géométriques à partir des squelettes et l'instanciation efficace des sous structures. Les développements autour de la programmation GPU sont également exploitables pour des visualisations fonctionnelles et constitue un premier axe de travaux à venir, certes technique, mais important.

Générer la géométrie avec le niveau de détail requis, au moment requis, requiert de revoir le codage des sorties de simulations. Nous explorons actuellement une nouvelle approche basée sur le graphe de simulation qui semble prometteuse car capable de reconstruire la géométrie des structures à tout âge passé. Les développements de cette approche constituent un deuxième volet important de recherche, certainement le plus important à court et moyen terme.

Sur le plongement géométrique du fonctionnement

Nous n'avons accordé que très peu d'efforts à ce jour sur la définition de géométries à partir du seul fonctionnement de la plante ou d'une parcelle. Cet axe mérite d'être creusé, et forme le troisième volet de recherche. C'est un axe comportant de grandes difficultés, la définition de géométrie sous contraintes (dynamiques) étant un axe de recherche classiquement difficile. Mais la dynamique de la mise en place de la production dans les plantes et les approches statistiques explorées dans nos expériences en imagerie médicale nous fournissent des pistes intéressantes à court terme. Le partenariat avec le LIAMA sur la reconstruction 3D à partir de scanner devrait également contribuer à cette problématique, selon un éclairage dual, celui de la caractérisation des couronnes.

Sur la simulation de la dynamique des paysages et leur visualisation

Dans le chapitre précédent, nous avons précisé que l'évolution des travaux consiste d'une part à revoir l'expression de modèles simulant la croissance – ce qui est en cours sous la conduite des modélisateurs, et, d'autre part, de contribuer, au sein d'une communauté élargie, à l'émergence de modèles conceptuels et d'environnements techniques pour la modélisation et la simulation d'écosystèmes. Il semble que des environnements de simulations implémentant le formalisme DEVS puissent répondre à cette attente, ce que nous évaluerons, sans que cela puisse, du moins à cette heure, constituer un axe de recherche en soit pour la visualisation et la représentation des

paysages. Par contre, la nécessité, sur des projets d'écologie, de disposer de multiples vues d'un système complexe provenant de multiples modalités (plantes réelles, reconstruites, simulées...) nous conduit à souligner l'importance des développements et de l'intégration par et pour le WEB, axe technique pour lequel le partenariat devrait se consolider.

Nous détaillons ici les grandes lignes de ces axes, illustrées par quelques résultats préliminaires obtenus récemment.

L'emploi des techniques GPU.

L'emploi du potentiel de la GPU est aujourd'hui classique en visualisation et se répand dans le calcul scientifique pour la simulation. Cependant, à ce jour, les technologies évoluent rapidement et les environnements de développements n'offrent pas encore une stabilité éprouvée. Dans ce contexte, et sans expérience technique importante, notre implication dans ce thème est à ce jour prudente, limitée, et conduite avec l'appui de partenariats.

La génération de la géométrie des plantes à partir de leur squelette.

On cherche ici à exploiter le potentiel de la GPU à générer de la géométrie des plantes à partir de primitives de plus haut niveau, compactes, et du squelette de leur structure en particulier.

Dans les modèles multi-échelles présentés, la représentation du système ramifié s'appuie sur un squelette de résolution variable avec un maillage adaptatif, généré à la volée lors du rendu. Il va de soit que la génération de ce maillage (une discrétisation de troncs de cônes) par la GPU serait efficace. Mais nous pouvons faire mieux.

Dans le cadre d'un partenariat avec la BJFU (Beijing Forestry University) nous remplaçons la génération du maillage décrivant les axes, actuellement effectuée par une polygonisation 3D par la CPU, par des primitives générées par la GPU en 2D $\frac{1}{2}$. La projection d'un tronc de cône peut en effet être représentée par deux quadrilatères à partir de deux nœuds du squelette avec leurs diamètres respectifs. Les normales sont simples à calculer dans l'espace image, et nous calculons aussi une profondeur pour chaque sommet généré afin de permettre un traitement satisfaisant des faces cachées par le Zbuffer. De manière analogue, nous utilisons la GPU pour générer un disque triangulé à partir d'un point et d'un rayon pour représenter une couronne simpliste. La GPU évalue les normales des points définissant le disque et modifie la profondeur du centre. Cette représentation simpliste peut être utilisée pour les paysages fonctionnels ou des représentations hiérarchiques telles que le E-squelette. Les premiers tests effectués sur des branches et des vues de paysages fonctionnels (couronnes) avec le laboratoire graphique de la BJFU sont encourageants (Figure Evol_1).

A cette heure, les tests menés ne couvrent pas d'aspects multi-échelles, mais l'algorithmique l'autorise aisément. En amont, en ignorant des nœuds sur les axes (concaténation à la volée), en aval en adaptant la complexité géométrique au diamètre apparent de l'axe ou de la couronne. Les problèmes d'aliasing restant entiers sur les petites primitives, nous devrions également développer des rendus dégradés sous formes de lignes et points, et disposer d'un mécanisme performant de choix et de chargement des différents moteurs de rendu dans la GPU.

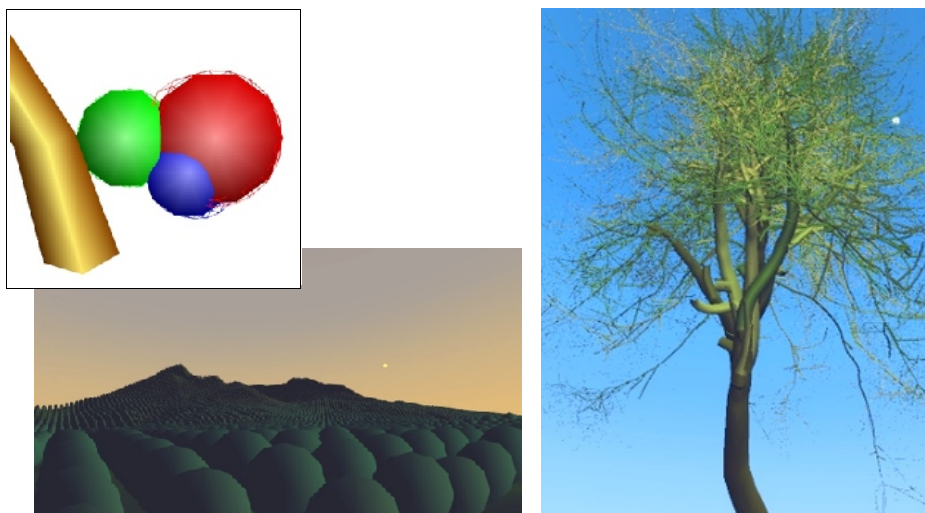


Figure Evol_1. Tests sur la génération GPU de primitives 2D 1/2 pour les branches et couronnes. (collection de l'auteur, en partenariat avec la BJFU, 2009-2010). A droite, branche de 3 noeuds, et 3 couronnes. A gauche, visualisation 2D 1/2 d'un chêne, les couleurs sont fonctions des diamètres. En bas: représentation par disques 2D 1/2 de biomasses cumulées dans un paysage fonctionnel.

A plus long terme, sur des représentations réalistes, des gains importants peuvent aussi être acquis si l'on choisit de travailler avec des textures d'écorces procédurales et non pas définies par des images. Ce type de texture est bien adapté à l'algorithmique que nous proposons et autorise la définition d'évolutions de textures le long d'un axe en fonction de l'âge des nœuds le définissant..

Ces techniques vont donc permettre une meilleure utilisation des approches multi-échelles présentées dans le chapitre 2, tout en restant opérationnelles sur de nouvelles approches de représentation de la structure des plantes, telles que décrites si dessous.

De l'usage des sous-structures.

L'exploitation des sous-structures topologiquement identiques dans les végétaux simulés par des implémentations du modèle GreenLab est une piste récurrente pour optimiser les performances en visualisation. Rappelons que la notion de sous-structure [MP_Yan003] est fondée sur le fait que toutes les arborescences de même âge chronologique et de même âge physiologique sont identiques. En fait, cela se vérifie si l'on ignore les effets mécaniques (torsion, ploiements,...) qui contribuent malheureusement grandement au réalisme visuel.

Pour la visualisation, contrairement à la simulation, l'instanciation des éléments doit être effective. Pire, l'exploitation des sous structures passe nécessairement par un post traitement de déformation hiérarchique, de préférence appliqué par la GPU sur chaque instance, en fonction de l'orientation de celle ci. C'est un développement qu'il est réaliste d'envisager, sur la base de transformations simples -localement linéaires telles que celles de Barr [CG_Barr984]. Ces transformations peuvent être prise en compte lors de la génération des primitives en GPU telle que présentée ci dessus. Une implémentation efficace de l'approche peut être obtenue si la représentation de la structure est effectuée axe par axe, des axes principaux vers les axes terminaux. Ce type de parcours de la structure est par ailleurs bien adapté aux représentations multi-échelle, et est compatible avec la génération des primitive géométriques décrite ci dessus. On peut donc envisager de calculer les ploiements et définir la géométrie lors des étapes ultimes du rendu, à la volée.

Les limites des approches multi-échelles et de l'instanciation.

A moyen terme, cependant, se pose fatalement la question du volume des représentations multi-échelles, car elles pénalisent l'interaction (les plantes sont statiques) et l'hétérogénéité. L'exploitation d'instanciations suppose par ailleurs que de nombreux individus identiques soient présents dans la scène avec les mêmes conditions de vues (même niveau de détail); hors ceci est rare, sauf dans des plantations homogènes. Il faudrait que l'on puisse mettre en évidence les structures identiques partagées par plusieurs plantes, de la même espèce mais pas du même âge. Ce qui revient à ne plus considérer la représentation de la culture ou du paysage par une approche arbre centrée, mais plus globale, voire fonctionnelle.

Une nouvelle approche pour créer des représentations géométriques

L'expérience acquise sur les modèles LOD montre que les structures géométriques sont in fine trop riches au sortir des simulations; beaucoup de ressources sont consacrées à dégrader l'information pour obtenir des représentations adaptées à la visualisation rapide. De surcroît l'exploitation d'instanciations clairement identifiées lors de la simulation s'avère lourde et incompatible avec des représentations polygonales usuelles. La littérature, bien qu'abondante sur la représentation des structures de plantes (voir le recueil relativement exhaustif chez [MP_Godin000]), traite peu de ces instanciations, et attache volontiers des attributs géométriques aux structures.

Pourquoi ne pas reléguer la génération de la géométrie le plus tard possible ? Et si cela s'avère possible, quelles sorties de simulation doivent être considérées sans compromettre la généricité de l'approche ?

Ces questions sont abordées depuis peu au sein de l'équipe DigiPlante sur un nouveau moteur de simulation GreenLab (dit "continu") [MP_Li010] et en partenariat avec l'Université de Tongji, travaillant sur la reconstruction par L_systèmes de plantes à partir de photographies [MRP_Sun009], [MP_Sun009].

Nous montrons en effet que l'on peut coder efficacement la topologie issue d'une simulation par un simple chainage des nœuds du squelette avec une structure simple de graphe (une seule instance de sous-structure est codée).

Pour chaque nœud, nous codons simplement ses successeurs en distinguant celui du même axe des ramifications. En présence d'une sous-structure on pointe sur la première instance de celle-ci. Dans ce codage, à chaque nœud, nous associons un attribut typologique (l'âge de différenciation ou physiologique), la date d'apparition, et la production (ou taille géométrique) du bois et des organes associés.

Ce codage est minimaliste mais complet: il est employé dans les simulations de nos moteurs de croissance structure-fonction pour permettre des arrêts - reprises de simulation (stop & go). Nous avons montré qu'un tel graphe peut être construit par tout modèle structure ou structure fonction en l'illustrant sur le modèle GreenLab continu, les L_systèmes [MP_Jaeger010] puis très récemment sur l'implémentation du modèle GreenLab usuel.

La reconstruction de la géométrie à partir du graphe de simulation

A partir de cette structure de donnée que l'on appelle le graphe de simulation, on peut reconstruire la géométrie, non seulement à l'étape finale de simulation, mais aussi à toutes les étapes antérieures si

l'on a prend soin de ne pas élaguer les nœuds morts de la structure. Le codage géométrique des organes est inutile, seul leur biomasse est nécessaire. Les informations de date permettent également de simuler les auto-élagages et définir des représentations différentes selon l'âge relatif des organes [MP_Jaeger010] (Figure LoD_20)

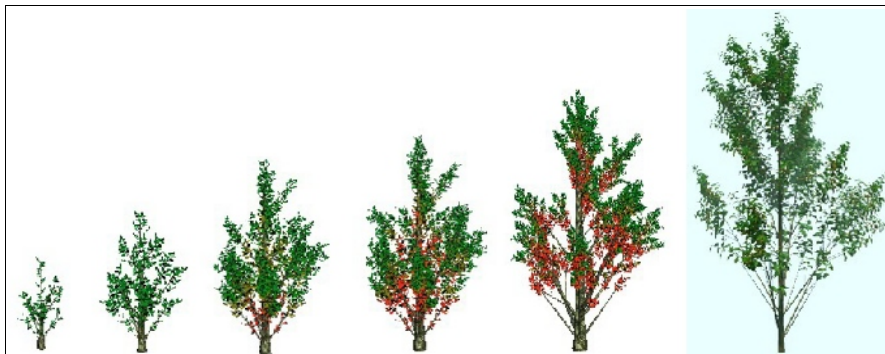


Figure Evol_2 Reconstruction géométrique de 6 étapes de croissance à partir d'une unique simulation. Les feuilles anciennes sont colorées avant de disparaître. La représentation réaliste, à droite, utilise une allométrie différente (issu de[MP_Jaeger010])

Ce codage est très performant (quelques Ko pour l'exemple illustré) et particulièrement adapté aux échanges sur le Web.

Du point de vue technique, il reste cependant de nombreux points d'amélioration dont plusieurs volets ont déjà été exprimés au préalable: le point premier est de pouvoir restituer des déformations géométriques liées au positionnement et à l'orientation des axes. Il s'agit en particulier des effets mécaniques. Elle peut être opérée classiquement sur la structure reconstruite, en perdant donc le bénéfice des instanciations liées aux sous-structures.

Nous opterons plus volontiers sur des travaux impliquant la GPU. Le programme de reconstruction géométrique est suffisamment simple pour envisager que la reconstruction soit totalement effectuée par la GPU. Elle pourra ensuite, de fait, intégrer des déformations mécaniques sur les instances de sous-structures.

Les acquis des modèles LOD décrits dans le deuxième chapitre pourront être exploités: en particulier la fusion de nœuds du graphe.

Enfin, au niveau du rendu, l'approche est totalement compatible avec les développements envisagés sur la visualisation GPU à partir de squelette (voir plus loin). Et de nouveaux développements spécifiques sont à explorer: la génération de textures exploitant les notions d'âges des nœuds du graphe, et bien sûr la minimisation de la reconstruction: le parcours du graphe, ordre par ordre, est bien adapté à une représentation multi-échelle. Par ailleurs, les occlusions (si elles sont rapides à estimer) dans les scènes n'imposeront pas forcément de générer la géométrie d'un axe ou d'une sous-structure.

La reconstruction de géométries d'étapes de croissances passées

Dans nos premiers développements, pour une date donnée, la reconstruction parcourt le graphe depuis la base du tronc (c'est à dire la graine), axe par axe, et ne restitue que les portions d'axe existantes à la date donnée. L'unique difficulté réside en fait dans la détermination de la taille

géométrique des organes. Celle-ci n'est en effet connue (en fait calculée à partir du volume connu de biomasse) qu'à la dernière date de simulation. Pour les nœuds des branches, on dispose du volume de moelle (stable) et des cernes de l'entrenœud sous-jacent. Le volume de moelle nous permet de connaître la longueur de l'entrenœud (que l'on considérera donc comme stable) et le diamètre initial. Le volume des cernes ajouté à celui de la moelle nous permet de définir le diamètre à l'âge final. Pour restituer le diamètre à un âge passé nous effectuons une interpolation. Sur les organes (feuilles) on considère que leurs expansions furent immédiates.

La reconstruction géométrique de la plante est donc une restitution exacte pour l'âge final de la plante, c'est à dire à la date de génération du graphe; elle est une interpolation (sur les diamètres et taille de feuilles/fleurs/fruits) pour les étapes passées. Définir une géométrie exacte, c'est à dire identique à celle obtenue par une reconstruction à chaque étape de croissance est possible théoriquement: il faut effectuer des dé-convolutions des productions (bois, feuille, etc) sur la durée de croissance des organes en question, dé-convolutions qui tiennent compte des facteurs environnementaux. Ce problème est donc un cas particulier -certes simplifié- d'identification.

Du point de vue visuel, cette dé-convolution n'est pas forcément une nécessité, car si l'élongation des axes est rapide (en un cycle, ce qui est généralement le cas), les longueurs des axes sont figées, la forme globale et locale de la plante est respectée, les erreurs que l'on pourra constater ne concernent que des diamètres locaux et les tailles d'organes. Elles seront relativement faibles et totalement bornées sur les diamètres des axes. Pour les feuilles ces erreurs seront très limitées également, car n'intervenant que si celles-ci sont présentes sur plusieurs cycles; et, dans ce cas, un autre facteur devient visuellement (fonctionnellement) plus important, celui de l'apparence, du vieillissement (avec des évolutions de couleurs en particulier).

En conclusion, l'exploitation d'un unique graphe de simulation calculé au stade final de croissance pour reconstruire l'ensemble des géométries passées de l'arbre se justifie totalement pour la visualisation, y compris dans des contextes d'applications forestières et agronomiques.

Reconstruire une culture, une forêt, un paysage à partir d'un unique individu ?

Il est bien évidemment facile de construire des scènes avec de nombreuses instances d'une unique représentation géométrique.

Dans les applications agronomiques et forestières on cherche à modéliser et représenter la variabilité constatée sur le terrain, celles qui paraissent « intra-spécifiques », celles liées à la compétition et aux conditions environnementales, celles -plus évidentes- liées à l'âge des individus. La question que nous posons ici est donc plus ambitieuse, pouvons-nous construire des scènes avec de multiples reconstructions géométriques distinctes -voir toutes distinctes les unes des autres- à partir d'un unique graphe de simulation pour représenter cette variabilité ?

Sur la représentation de la variabilité intra-spécifique.

Dans les moteurs de croissances cette variabilité est explorée et simulée avec deux approches distinctes: l'une basée sur la représentation moyenne et sa variance – dans ce cas on va rarement jusqu'au calcul de la structure, l'autre basée sur une simulation stochastique. Dans GreenLab, cette simulation est effectuée par des représentations multiples et stochastiques des sous-structures [MP_deReffye003a]. Dans la simulation on choisit l'un des représentants parmi la distribution pré-calculée (on montre que le volume de ces distributions n'est pas exponentiel, et que l'on arrive à construire des plantes avec un nombre constant de représentant dans chaque classe). Mais à ce jour, en sortie dans le graphe de simulation, aucune information spécifique n'est fournie dans une simulation présentant une nature stochastique (elle est implicite pour les sous-structures sélectionnées). Il va donc falloir définir les sous-graphes des distributions, puis spécifier dans le

graphe les nœuds où les choix de représentants sont à faire pour reconstruire une plante en particulier. Moyennant quoi la simulation de cette variabilité sera effective.

Représentation d'une scène composée d'une même espèce avec des individus d'âge différents.

Nous avons vu que le graphe de simulation permettait d'effectuer des représentations géométriques passées de la plante simulée. Nous pouvons penser générer une forêt composée d'individus d'âge distincts d'une même espèce à partir d'une unique simulation, en instanciant (avec ou sans variabilité intra-spécifique) les reconstructions passées. Oui, mais celles-ci correspondent à des dates passées, donc à des conditions environnementales différentes, et donc avec des volumes de biomasse différents... et donc à des structures (ou du moins des tailles) différentes. Ceci dit, dans de très nombreuses applications (aménagement du territoire, jeux, ...), nous pouvons considérer cet environnement comme constant et donc générer des forêts ou cultures composées d'individus d'âge différents à partir d'une unique simulation du plus vieux d'entre eux. Ce n'est bien sûr plus le cas si l'on désire représenter l'effet de compétitions ou de variations de ressources environnementales..

La représentation de la plasticité (compétition, changements environnementaux)

La restitution de la plasticité de la structure en fonction de l'environnement est certainement la difficulté et l'enjeu majeur de cette approche. Peut-on reconstruire les structures et géométries des plantes dans des contextes de compétitions à partir d'une unique simulation ?

Dans le cas de plantes agronomiques simples (sans rétro-action de la production sur la structure, non ramifiées), on peut considérer que la topologie reste stable avec des conditions environnementales différentes. Dans ce cas, nous sommes donc en mesure de représenter fidèlement l'hétérogénéité due à la compétition dans une culture à partir d'un unique individu. Il faudra cependant ré-évaluer la production et changer les tailles des organes en conséquent.

Pour les arbres et arbustes, la rétro-action est en général importante (voir chapitre précédant simuler et représenter la compétition, en Figure Pf_1).

La représentation de la plasticité pose, dans le cas général, trois types de difficultés.

La première est de disposer pour un nœud donné, des sous-graphes de simulation qui pourront potentiellement être fils de ce nœud et de choisir un bon représentant. Ce point est le plus simple à réaliser, c'est une extension du cas stochastique, où, au lieu de disposer d'une distribution de sous-graphes équivalents, nous disposons d'une classification de sous-graphes valués par un critère de coût (dans le modèle GreenLab cette plasticité est guidée par un ratio particulier nommé offre sur demande ... c'est à dire biomasse disponible sur biomasse requise pour l'établissement de la structure. Cette grandeur est calculable sans implémenter un modèle spécifique dans la reconstruction). Si l'on se contente d'applications strictement inscrites dans le contexte multimédia, une variante serait d'éviter de construire ces sous-graphes en utilisant une unique simulation d'une plante «complète, idéale» calculée sans compétition et limitation de ressources. On opère ensuite des parcours partiels stochastiques du graphe de simulation pour s'inspirer des réalisations partielles de la structure.

Le deuxième point difficile est que les attributs des nœuds (la production) doivent être réactualisés en fonction des sous-graphes sélectionnés car ils modifient le niveau de production et de demande. Sur ce point, pas de mystère, il faut revenir à la simulation. Car le système est dynamique; le niveau de production va conditionner la structure du prochain cycle.

Enfin, et c'est un point difficile, il faut calculer/ré-évaluer les conditions environnementales qui influent la production: ce qui revient à faire de la simulation de paysage fonctionnel. Dans

l'expérimentation présentée sur la compétition (Figure Pf_1), elle est obtenue par le diagramme de Voronoï. Or, tant que les plants sont jeunes, cette compétition spatiale ne s'applique pas, et, réciproquement, il y a un effet «d'emballlement» sur les sujets matures, quand les surfaces de projection se télescopent. Le diagramme qui définit les classes de compétition sont donc à réévaluer à chaque étape, ou, ce qui, serait plus judicieux, à exploiter des formalismes de pavage plus complexes capables de prendre en compte des pondérations paramétriques.

Dans ce dernier cas nous ne sommes bien sûr plus sur un problème de représentation/visualisation. On a donc clairement défini les limites et les bornes de cette approche de graphe de simulation. Mais cette complexité croissante d'usage de la représentation propose un cadre d'études et de développement et montre que l'on peut progresser efficacement sur la représentation des scènes avec un réalisme comportemental simpliste mais réel... et avec un coût de complexité géométrique très faible.

Vers une représentation géométrique de la production des plantes

Une manière efficace de s'affranchir de la complexité de la représentation des plantes virtuelles est de construire celle-ci indépendamment de sa structure, sur la seule base de sa production (volumes de biomasse de bois et de feuilles).

A cette heure peu de travaux ont été effectués dans ce sens. Ils sont aujourd'hui limités à la représentation de la biomasse cumulée, soit plongée selon le squelette des plantes dans un espace voxel, soit plongée sous une forme simple (sphère).

On peut envisager d'autres représentations équivalentes. Si l'on considère la forme de la couronne comme représentative de la masse foliaire, l'équivalence peut être définie selon le volume (à densité de feuille constante), ou selon la surface (la surface foliaire elle même). Nous avons conduit une première étude menée dans le cadre d'un stage de Mastère (Charles Antoine Douet et Murielle Luong de l'Ecole Centrale de Nantes). Dans cette étude, les formes de couronne proposées sont celles utilisées par la communauté scientifique en foresterie, formes auxquelles se rattachent des modèles d'évolution et de qualité des bois.

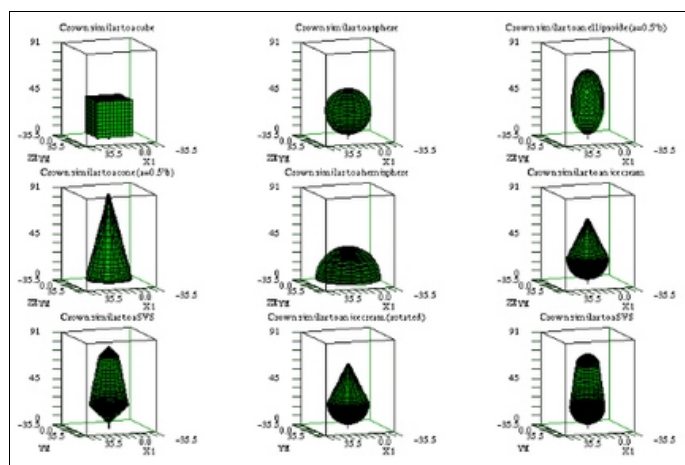


Figure Evol_3. Étude sur des représentations équivalentes de couronnes à surface constante (C.A. Douet, M. Jaeger, 2006)

Cette approche peut devenir particulièrement efficace dans le cas où l'on utilise un modèle structure

fonction tel que GreenLab. Grâce aux sous-structures inhérentes au modèle, nous pouvons attacher des représentations multi-échelles à la production. Et faire bien sur le pont avec l'approche statistique des E-squelettes sur des représentations structurelles grâce à l'étude de la dispersion spatiale de la production.

La caractérisation des formes des houppiers à partir de leur contenu est donc une piste intéressante pour effectuer le pont entre les représentations structurelles et fonctionnelles. De plus, grâce au partenariat avec le LIAMA et le LERFOB, la caractérisation des couronnes peut également être étudiées à partir des acquisitions lasers sur des plantes réelles.

La définition géométrique par la production peut également être reliée à des études théoriques intéressantes sur les stratégies d'occupation de l'espace, c'est à dire sur les modèles architecturaux. Supposons en effet que la distribution des feuilles soit proche de l'enveloppe de la couronne. Selon la nature de l'éclairage (diffusion importante -pays du nord versus illumination directe – tropiques), les représentations des couronnes ne seront pas équivalentes.

Si l'on considère de plus le résultat de l'organogénèse (mais sans plongement de la structure), on disposera à chaque étape de croissance, d'un dispositif de contraintes important. A un stade de croissance donné, nous connaissons la distribution du nombre de feuilles pour chaque stade (classe d'âge) avec leurs surfaces respectives, idem pour les entre-nœuds. Si l'on exploite la dynamique de ces distributions, on devrait pouvoir en déduire des éléments structuraux et de positionnements géométriques. Par exemple si le nombre d'entre-nœuds est proportionnel à l'âge, nous sommes en présence d'un modèle de corner (comme pour un palmier). Dans ce cas on pourra également définir de manière explicite et déterministe la longueur et le diamètre de chaque entre-nœud, et donc la position de chaque feuille. A terme, ce type d'approche devrait permettre de définir non seulement des géométries de couronnes «équivalentes» mais des structures équivalentes.

L'intégration WEB. Le futur des applications de recherche sur l'environnement ?

Avec le développement de la modélisation des systèmes complexes autour de l'environnement, le partage de la représentation de l'environnement devient crucial. De nos interactions avec nos collègues hydrologues, écologues, économistes, ..., émerge la nécessité, sur des projets de développement durable, de disposer de multiples vues d'un système complexe provenant de multiples modalités (plantes réelles, reconstruites, simulées...). En effet, les représentations d'un même site, d'un même objet d'étude (le bassin versant, le paysage) sont partielles et partiales, masquant des approches, des outils et des méthodologies différentes, voire incompatibles. Les développements et l'intégration des outils de visualisation par et pour le WEB sont donc des moyens importants pour construire le dialogue interdisciplinaire. Cet axe « intégration Web pour la représentation des paysages », que nous qualifions de technique, aura une importance croissante pour la valorisation des modèles et le développement des partenariats. On ne peut travailler sur la représentation des plantes ou paysages sans prévoir à moyen terme de restitutions sur des supports déportés; les visualisations via le WEB sont ouvertes à un large public, et surtout elles peuvent être rattachées à des sites d'études totalement identifiés avec des fonctions de navigation géo-référencées.

La maîtrise des outils WEB et le développement d'application de représentation du végétal sur le WEB relève cependant de compétences à acquérir; ce que nous cherchons à développer via des partenariats en local (École IMERIR sur Perpignan) et à l'international (Université de Tongji) dans

le cadre de projets appliqués.

Ainsi, en partenariat avec le LIAMA et l'Université de Tongji, nous participons au développement d'une plateforme technique visant à intégrer des représentation réelles et virtuel d'un site de recherche en zone écologique: l'île de ChongMing [CG_Jaeger010] (Figure Evol_4). Dans ce cadre nous cherchons plus précisément à construire de nouvelles modalités de représentation compacte des plantes.

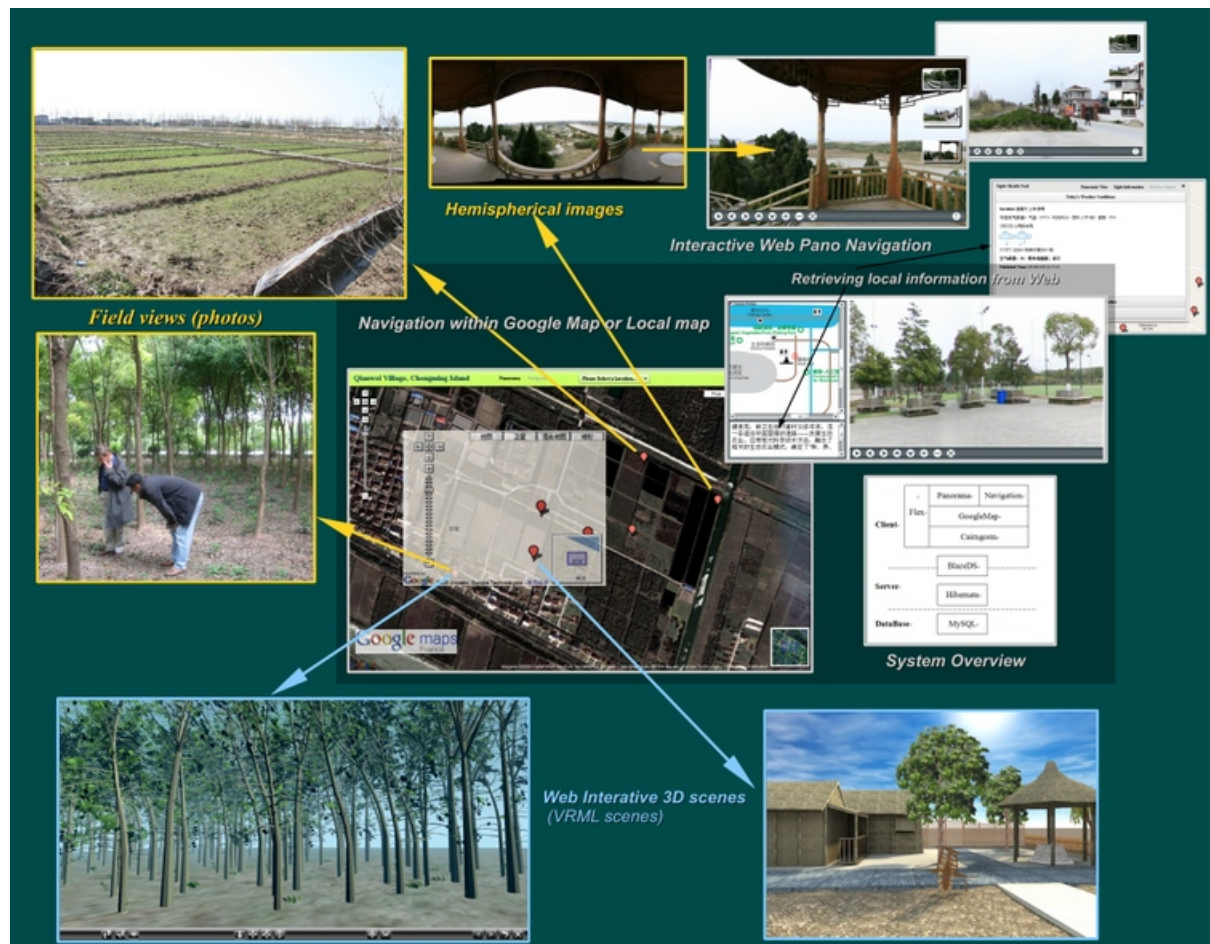


Figure Evol_4. Vues réelles et simulées depuis Google Map (issu de [CG_Jaeger010]). Des vues satellite et navigation, on accède à des cartes descriptives locales, des vues panoramiques, des vues fixes, des scènes 3D en script VRML, et des requêtes locales (météorologiques ici).

La représentation de scènes naturelles sur le Web est soumise aux contraintes classiques de la visualisation temps réel et à celles de l'usage des postes clients: la limitation de la bande passante, la portabilité des applications de restitution sur du matériel hétérogène aux performances graphiques variables. Les travaux de compression / décimation de la géométrie peuvent donc être valorisés dans ce cadre.

A ce jour, nous avons simplement exploité un certain nombre de représentations géométriques peu gourmandes pour alimenter les points d'intérêt du site. Ces représentations sont des représentations de bas niveau LOD polygonales et images telles que décrites dans le chapitre 2.

Nous avons d'autre part développé des encapsulages de générateurs de végétaux (un mini générateur L_systems) dans des proto JavaScript VRML. Les règles, le générateur et les paramètres sont

envoyés au client, qui génère et affiche la géométrie de la plante [MP_Sun009]. Ces résultats, marginaux et partiels, sont cependant extrêmement prometteurs si l'on considère l'ensemble des développements en cours.

A court terme, aux règles et au générateur de L_systèmes peuvent être substitué la structure simple de graphe topologique issu d'un simulateur, et le programme de reconstruction géométrique tel que décrit plus haut. Reste à envoyer au client la description de la scène, de préférence sous la forme d'un biotope. Sur le client, en fonction du point de vue, les positions des éléments végétaux sont calculés avec leurs âges respectifs. Pour chaque plante visible, on reconstruit ensuite une représentation correspondant à l'âge de l'individu avec le niveau de détail adéquat.

A moyen terme, et avec les réserves émises sur l'usage du graphe de la structure simulée, on peut lier les représentations de la reconstruction à des paramètres climatiques environnementaux moyens, liés au site (température, précipitations).

A long terme, c'est pour l'ensemble de la communauté scientifique intéressée les développements sur les paysages dits fonctionnels que cette technologie est intéressante. Notre petite expérience sur la simulation bio-physique (aspects biologie végétale et aspects hydriques) nous a enseigné qu'il était difficile voire impossible d'acquérir les compétences pour développer conjointement en une même équipe et sur un même site des modèles à mobiliser ensemble sur des simulations de systèmes complexes. Avec des formalismes partagés sur l'accès aux données (tels que DEVS), on peut envisager, des simulations dont les composants modèles, les données, les paramètres seront distribués, à l'image de travaux engagés sur des systèmes multi-agents [MSL_Campos998].

CONCLUSION

L'un des grands enseignements des récentes rencontres scientifiques dédiés à la modélisation des plantes est que les modèles de simulation de la croissance des plantes dits structure fonction (FSPM) arrivent aujourd'hui à maturité. Il aura fallu plus de 20 ans pour que les approches structurales décrivant l'architecture des plantes et les approches fonctionnelles centrées sur la modélisation de la production se rejoignent. Les deux composantes structure et fonction s'équilibrent aujourd'hui. Cette maturité se constate par l'existence de formalismes et surtout par l'usage de ces modèles. Les travaux portent peu aujourd'hui sur ces modèles proprement dits, mais se focalisent sur des questions, sur des applications précises pour lesquels les FSPM donnent un cadre, un support. L'heure n'est plus à montrer le potentiel de ces approches mais à les utiliser pour tenter de répondre à une multitude de problématiques.

Cela fait plus de 25 ans que l'on a vu pour la première fois un arbre virtuel sur un écran couleur. Aujourd'hui la visualisation des plantes virtuelles tant dans les contextes agronomiques que multimédias a également atteint un stade de maturité. Il n'y a pas d'équipe de recherche intégrant l'architecture des plantes qui ne soit en mesure de proposer des représentations tridimensionnelles de celle-ci. Dans les projets d'aménagements comme les jeux vidéo les scènes présentent des environnements naturels au réalisme stupéfiant, de surcroît dans des environnements interactifs.

Que de progrès dira-t-on !

Mais entre les images de plantes simulées par les agronomes -uniquement par des modèles structuraux- dans la fin des années 80 à celles générées aujourd'hui, il est bien difficile de dater et de faire un tri. Et du Mighty Maple de Bloomenthal aux arbres hybrides dans les jeux scènes de Crytek, où sont les innovations ? Pas dans la précision de la description géométrique des plantes, bien au contraire, elle s'est appauvrie par l'usage massif d'imposteurs.

En fait, dans la représentation des plantes virtuelles, on a surtout gagné en efficacité, on visualise rapidement, avec interactivité dit-on. Et on peut le faire car non seulement le matériel le permet, mais surtout parce que la structure de l'objet est transformée, simplifiée. De la structure on tend vers la forme – au sens littéral du terme, synonyme d'apparence. Et l'apparence dépend du contexte, de la distance à l'objet... ce qui peut expliquer qu'il n'y a pas une méthode efficace multi-échelle pour représenter ces structures mais bien une panoplie. Toute la difficulté réside dans l'enchaînement des approches, point crucial dans le contexte multimédia.

Doit-on donc considérer qu'il n'y a plus de problématique de recherche sur la visualisation des plantes ? La réponse est oui si l'on attend de cette visualisation qu'elle reste un décor, un contexte.

Ce n'est évidemment pas ce qu'attendent les communautés intéressées par le monde végétal -qu'elles soient de recherche ou non. Elles ont besoin d'appréhender, de comprendre des dynamiques. Ni même la communauté de l'infographie qui a besoin «d'objets interactifs», interagissant avec un utilisateur. La plante virtuelle devient donc enfin un objet vivant, animé. Ce qui ne permet plus des simplifications de structures figeant le végétal comme un décor de théâtre.

Mais l'élément déterminant de l'évolution de la représentation des scènes naturelles n'est pas cette attente. Il réside simplement dans le fait que les applications des modèles structures fonctions et les besoins du multimédia ne portent pas sur la plante isolée mais sur un ensemble de plantes, des cultures, des forêts, des paysages virtuels. Et dans ces scènes il n'y a aucun intérêt à représenter la même plante à perte de vue... Mais la variabilité dans les scènes constitue un frein important que les évolutions techniques (mémoire et GPU) ne peuvent surmonter pour l'heure.

Et bien sûr, à ces échelles, le besoin d'appréhender, de comprendre les dynamiques est encore plus important, crucial même, qu'à celle de l'individu plante, il revêt d'enjeux sociétaux.

Dans cet exposé, nous avons essayé de poser des axes de réflexion pour proposer des outils de représentation de plante et paysages suffisamment robustes et efficaces pour permettre d'appréhender les variabilités et les dynamiques de plantes, cultures et paysages simulés.

Lorsque l'on considère les approches de visualisation de plantes, on suit une trame claire, unique dans le cheminement allant de la simulation (la structure) vers l'apparence. Celle de la réduction de la complexité par agrégation de composants. Ce que l'on attendrait d'un modèle efficace est exactement l'inverse: disposer d'une géométrie simple, que l'on raffine en fonction des besoins et contraintes de la visualisation.

C'est la manière dont nous percevons notre environnement. Et ce que percevons de premier abord, ce n'est pas une forme. Pour toute civilisation sur notre terre, l'apparence de l'environnement végétal est avant tout une couleur: le vert. C'est le vert de la production, celui de la fonction. Il y a donc une autre trame à exploiter, celle de la prise en compte de la fonction, c'est à dire de la production végétale.

S'il faut reconnaître que l'exploitation de cette trame est marginale jusqu'ici, elle n'en reste pas moins prometteuse. C'est une grandeur scalaire, dynamique, que les modèles peuvent évaluer de manière hiérarchique, rattachée ou non à une définition explicite de la structure. La biologie nous apprend que les assimilats disponibles pour établir ou faire croître les organes forment un pool commun, et donc que la localisation -le positionnement géométrique ou topologique, voire la nature de la structure topologique- n'influent pas leur distribution mais conditionnent le nombre d'organes à venir. Il n'y a donc pas besoin d'effectuer des plongements géométriques pour simuler et caractériser la dynamique du système. Du point de vue fonctionnel, ce qui importe est simplement les distributions d'organes par classe typologique et tranche d'âge avec leurs biomasses respectives. Mais si l'on veut percevoir le système il faut lui attribuer une structure et un plongement géométrique et ces opérations sont coûteuses de part leur nature même.

On a donc tout intérêt à instancier la structure et effectuer des plongements géométriques le plus tard possible dans la chaîne de visualisation et bien sur le plus efficacement/rapidement possible.

Repousser la reconstruction géométrique et l'instanciation de la structure des plantes de la simulation vers le rendu définit donc aujourd'hui le premier axe important de recherche et développement.

En ne générant pas explicitement la structure, en ne calculant pas la géométrie lors de la simulation, nous montrons que l'on peut exploiter un graphe de simulation pour générer de multiples représentations. Nous avons souligné que la connaissance des biomasses en chaque nœud permet de reconstruire la géométrie de la plante et ses états passés. Et elle est indispensable pour simuler la plasticité de la structure et de la taille des organes dans les dynamiques complexes liant la plante à son environnement. Mais à cette heure, les travaux engagés ne forment qu'une piste, certes sérieuse mais confrontée à de multiples questions. Deux classes de problèmes émergent: la première concerne le réel potentiel de l'approche: jusqu'où peut-on aller dans la plasticité des reconstructions à partir des graphes de simulation sans que ces dernières ne deviennent elles même de lourdes simulations ? La deuxième concerne la définition géométrique et son réalisme: comment peut-on exploiter efficacement les graphes de simulation pour construire rapidement des géométries soumises à des contraintes d'orientation et de positionnement (aspects mécaniques notamment) ? Comment obtenir des rendus réalistes prenant en compte l'âge des nœuds ? Cette deuxième classe

de problèmes est sans aucun doute plus facile à prendre en compte, focalisée sur les techniques de visualisation, c'est à dire de rendu, et donc plus proche de la technique, et dont on pense que l'exploitation des capacités de la programmation GPU sera importante.

Partir de la production (la fonction) pour aboutir au plongement géométrique (la forme) est un cheminement dual de l'approche usuelle et forme le deuxième grand axe de réflexion sur la représentation du monde végétal.

La définition de représentations géométriques de la production est indéniablement à construire. C'est aussi un problème insuffisamment contraint, ouvert. Ce que nous proposons est de travailler sur des formes élémentaires, simples, pour lesquelles les volumes, les surfaces d'enveloppes sont proportionnels aux productions. En exploitant la dynamique des productions puis le nombre de composants dans la plante on espère contraindre la définition de ces formes. Et bien sûr on se posera la question du réel potentiel de l'approche: jusqu'où peut -on aller dans la la définition des contraintes à partir de l'exploration de la dynamique sans que ces dernières ne deviennent elles même des approches d'identification proches de celles de la calibration des modèles de simulation ? Et, par mimétisme à l'approche structurelle, quel degré de réalisme pouvons nous espérer par des approches aussi simplistes ? Pour quelles performances ? Pouvons nous y attacher des techniques efficaces -telles que celles de texels- décrivant la distribution interne des organes ? Là aussi, cette classe de problèmes semble plus simple à appréhender, car également relative au rendu, et dont on attend aussi un apport déterminant de la programmation GPU.

Dans ces deux cheminements oscillants maintenant entre structure et fonction le plongement géométrique (la forme) devient une expression. Et nous espérons que ces cheminements viendront à se croiser. La forme est le support par lequel les comparaisons peuvent êtres confrontées et comparées. L'actualisation des études statistiques de la dispersion géométrique (nuées dynamique et E-squelettes) est une piste à revisiter dans ce sens. D'autant plus qu'aujourd'hui les sous-structures, respectivement les cohortes d'organes, forment des classes d'analyse organisées selon la structure, respectivement selon la fonction.

Il est un élément d'importance à considérer lorsque l'on effectue des travaux de visualisation. C'est celui de la technologie. Et deux volets distincts mais non disjoints sont à maîtriser: les techniques GPU et les techniques WEB. Ces volets techniques sont orthogonaux aux axes de réflexions qu'ils contribuent à renforcer. Dans notre contexte de recherche, orienté sur la biologie, la maîtrise de ces compétences et technologies passe nécessairement par le partenariat.

L'exploration des techniques GPU est un axe technique pour lequel trois types distincts de travaux sont à considérer à relativement court terme.

Le premier est relatif à la génération de primitives géométriques. Nous avons en particulier évoqué la génération de représentations 2D $\frac{1}{2}$ pour représenter un maillage à partir d'un squelette. Elle s'opère de même pour des formes «simples» caractérisant le feuillage à partir d'approches fonctionnelles et devrait pouvoir s'étendre à des formes plus complexes, fonctionnellement équivalentes.

Le deuxième volet relève de déformations non linéaires en particulier de simulation de flambage sur des structures hiérarchiques. Ce type de transformation est classiquement homomorphe à des courbes cubiques, mais serait intéressant à explorer avec des transformations de Barr, bien adaptées au contexte GPU.

Le troisième volet est relatif au réalisme visuel. Il s'agit de définir des fonction de matériaux,

textures, indexés sur les âges des composants lors du rendu, en privilégiant des approches procédurales.

A plus long terme, les enjeux, s'inscrivent dans l'énoncé des deux axes de recherche précités. L'évolution de l'accès aux ressources de programmation des cartes graphiques, avec la banalisation d'environnements de calculs génériques tels que CUDA, conduira vraisemblablement ces problématiques vers des approches plus proches de la représentation, voire de la modélisation.

Enfin les techniques WEB nous imposent de penser et réfléchir différemment la représentation géométrique et la visualisation de l'environnement naturel. La visualisation de demain sera collaborative et distribuée et vraisemblablement obtenue par des modules de simulation eux mêmes distribués. Nos développements doivent donc être considérés comme des applications clients-serveurs avec des contraintes de performance fortes en terme de bande passante. Les axes de recherche présentés s'approprient bien à ce type d'environnement. En exploitant des représentations de plus haut niveau (graphe de simulation, résultat de la production), on minimise la bande passante tout en augmentant la souplesse des restitutions.

C'est indéniablement dans ce contexte que nous devons positionner nos travaux sur la simulation et la représentation des paysages fonctionnels. Notre petite expérience sur cette thématique nous a fait prendre conscience que nous devons penser nos approches comme des constituants de grands puzzles de systèmes complexes. Et donc de revisiter nos approches et outils pour qu'ils soient en mesure de devenir des pièces de puzzle. Ils doivent pouvoir constituer des composants fiables de ces systèmes. Cela passera inmanquablement par une formalisation de nos modèles explicitant clairement leurs comportements sur des cycles non synchrones avec leurs propres évolutions; par la définition d'interfaces étanches sur les entrées et sorties; et bien entendu par la définition précise des niveaux d'échelles et plus généralement des conditions de validité des approches et outils.

Ce ne sont donc pas seulement les représentations géométriques et visualisations des plantes et paysages virtuels que les techniques WEB nous imposent de repenser différemment mais toute notre production de recherche.

A nous d'œuvrer pour que nos outils méthodologiques, techniques, de modélisation, simulation et restitutions visuelles soit performants et ouverts au monde.

Index des figures

- p. 17. Int_1. Reconstruction d'un paysage du carbonifère (auteur: R. Lecoustre [MP_Davie000])*
- p. 19. Int_2. Les niveaux de représentation pour la visualisation des plantes et paysage (2003)*
- p. 24. Mod_1. Palmier avec ses formes pour l'habillage (issu de [MP_Jaeger992])*
- p. 25. Mod_2. Les sous-structures dans GreenLab (issu de [MP_Yan003])*
- p. 25. Mod_3. Palmeraie sous la brume (collection de l'auteur; 1992)*
- p. 26. Mod_4. Scène avec post-traitements (collection de l'auteur; 2006)*
- p. 27. Mod_5. Ellipsoïdes statistiques, valeurs d'occlusion (issu de [CG_JT007b])*
- p. 27. Mod_6. Rendu sans et avec occlusion ambiante (issu de [CG_JT007b])*
- p. 28. Mod_7. Reconstruction implicite à partir de squelette (issu de [CG_Yin005])*
- p. 30. Vol_1. Visualisation volumique (combinaison d'approches) (issu de [IV_Treil997])*
- p. 30. Vol_2. Visualisations volumiques par fonctions de transfert (issu de [IV_Nicolini992]).*
- p. 31. Vol_3. La caractérisation d'un solide par son E-squelette (issu de [IV_Banegas999a]).*
- p. 32. Vol_4. Applications des E-squelettes aux plantes virtuelles (collection de l'auteur; 2000).*
- p. 33. Vol_5. Construction du squelette hiérarchique (issu de [TIS_Zhang010]).*
- p. 34. Rec_1. Points laser et extraction des branches (issu de [MRP_Zhu010]).*
- p. 35. Rec_2. Clusters dans la couronne et reconstructions (issu de [MRP_Zhu010]).*
- p. 36. Rec_3. Extraction de branches sur acquisitions laser d'arbres isolés (issu de [TIS_Dai009])*
- p. 36. Rec_4. Données laser initiales et calculs des normales (issu de [MRP_Ning009])*
- p. 37. Rec_5. Classification dans une scène urbaine simple (issu de [MRP_Ning009])*
- p. 37. Rec_6. Segmentation d'un examen laser forestier (en soumission, 2010).*
- p. 40. LoD_1. Concaténation des nœuds à l'issue de simulations (issu de [MP_Jaeger987])*
- p. 40. LoD_2. La reconstruction du système ramifié (issu de [LOD_Zhang006b])*
- p. 41. LoD_3. Plongement géométrique multirésolution (issu de [LOD_Zhang006b])*
- p. 41. LoD_4. Performance des modèles LOD de branches (issu de [LOD_Deng010])*
- p. 42. LoD_5. Vue temps réel dans une peupleraie (collection de l'auteur , 2007)*
- p. 43. LoD_6. Union de 2 feuilles (issu de [LOD_Zhang006a])*
- p. 43. LoD_7. Union de feuilles dans une unité de croissance (issu de [LOD_Zhang006a])*
- p. 44. LoD_8. Usage de la densité pour l'union des feuilles (issu de [LOD_Deng010])*
- p. 44. LoD_9. Niveaux de détail selon l'orientation des axes (issu de [LOD_Deng007b])*
- p. 45. LoD_10. Conversion de l'arbre des unions de feuilles en table (issu de [LOD_Deng010])*
- p. 45. LoD_11. Performances en compression (issu de [LOD_Deng010])*
- p. 46. LoD_12. Performances sur une scène de forêt (issu de [LOD_Deng010])*
- p. 47. LoD_13. Modèles multi-échelles de feuilles et feuillage (issu de [LOD_Li010])*
- p. 48. LoD_14. Les formes "pixels" (issu de [MP_Jaeger987])*
- p. 49. LoD_15. Atlas de textures couronne et arbre entier (issu de [RRB_Bao010])*
- p. 49. LoD_16. Vue d'une forêt par nuage hiérarchique d'imposteurs (issu de [LOD_Deng007a])*
- p. 50. LoD_17. Plongement d'une scène dans un espace voxel (issu de [IV_Jaeger003])*
- p. 50. LoD_18. Vue réaliste d'une scène voxel (collection de l'auteur)*
- p. 51. LoD_19. Une scène voxel constituée de plus de 500 plantes ()*
- p. 55. Pf_1. Classes de compétition sur une forêt équienne. (issu de [MSL_Cournede010])*
- p. 55. Pf_2. Visualisation de scène avec compétition. (issu de [MSL_Cournede010])*
- p. 56. Pf_3. Premier prototype de simulation fonctionnel de paysage ([MSL_leChevalier0007a])*
- p. 57. Pf_4. Visualisation volumique de paysage simulé (collection de l'auteur; 2006)*
- p. 57. Pf_5 Visualisation d'un paysage fonctionnel simulé. (issus de [MSL_leChevalier0007a])*
- p. 58. Pf_6. Visualisation fonctionnelle d'un paysage(collection de l'auteur; 2008)*
- p. 59. Pf_7. Suivis des canaux dans un paysage fonctionnel ([MSL_leChevalier0007b]).*
- p. 59. Pf_8. Représentations réalistes d'un paysage fonctionnel (collection de l'auteur; 2009)*
- p. 60. Pf_9. Principe de synchronisation modèles-données (issu de [MSL_leChevalier2010b])*

- p. 61. Pf_10. Un conteneur pour la modélisation de la ressource (Vincent le Chevalier, 2010)*
- p. 61. Pf_11. Processus en compétitions pour des ressources (issu de [MSL_leChevalier010b])*
- p. 66. Evol_1. Tests sur la génération GPU de primitives 2D $\frac{1}{2}$ pour les branches et les couronnes*
- p. 68. Evol_2. Reconstruction géométrique de 6 étapes de croissance (issu de[MP_Jaeger010])*
- p. 71. Evol_3. Étude sur des représentations équivalentes de couronnes (C.A. Douet, M. Jaeger)*
- p. 73. Evol_4. Vues réelles et simulées depuis Google Map (issu de [CG_Jaeger010])*

Références

Note: afin de favoriser la lecture du document les références bibliographiques sont présentées sous la forme suivante XXX_NameYYY ou XXX est un acronyme clé correspondant au domaine de référence, et YYY les 3 derniers chiffre de l'année de parution.

Les acronymes et leurs domaines respectifs sont les suivants :

CG: (Computer Graphics) Synthèse d'image. Points techniques et généralités

MP: (Geometrical/Structural Plant Modeling) Modèle géométriques de plantes (par règles)

MRP: (Reconstructed Plant Models) Reconstruction 3D de plantes à partir d'images ou points laser

MPU: (User defined Plant Models) Modèle de Plante construits interactivement (Sketch based)

MSL: (Modelling the Structure of Landscapes) Systèmes Complexes, Modelisation de paysages et cultures

LOD: (Level Of Detail) Modèles et visualisation multi-résolution de plantes

RRB (Realistic Rendering -Billboards) Rendu realiste de vegetation par images (imposteurs)

IV: (Volume Imaging) Imagerie volumique et leurs applications

ST: (Software Tools) Outils logiciels, normes

TIS: (Image and signal processing) Traitement d'image – Signal

Synthèse d'image. Points techniques et généralités

[CG_Banegas999] Banégas F., Michelucci D., Roelens M., Jaeger M., 1999. An automatic adaptive surface reconstruction from ellipsoidal skeleton. In: 4th International workshop on implicit surfaces, J. Hugues and C. Schlick (Eds), Talence, France 13-15/09/1999. ACM Press; p. 113-122.

[CG_Barr984] A. Barr. Global and local deformations of solid primitives. In Computer Graphics, volume 18, pages 21–30. ACM Press, 1984.

[CG_Bloom990] Bloomenthal, J., Wyvill, B.: Interactive Techniques for Implicit Modeling. In: Computer Graphics Forum, Vol. 24. (1990) 109–116

[CG_Bloom991] Bloomenthal, J., Shoemake, K.: Convolution Surfaces. In: Proc.SIGGRAPH'91, Vol. 25. (1991) 251–256

[CG_Boisso984] J. D. Boissonnat. Geometric structures for three-dimensional shape representation. In ACM Transactions on Graphics, volume 3, pages 266–286. ACM Press, 1984.

[CG_Borienne993] Borienne Ph., Jaeger M., 1993. Représentation à base topologique sur un espace discret. In RL992: Colloque de Géométrie Discrète en Imagerie, Fondements et applications, Strasbourg (France), 20-21 sept. 1993, p. 11-22.

[CG_Bresen965] J. Bresenham. 1965. Algorithm for Computer Control of Digital Plotter. IBM System Journal, 4:25–30.

[CG_Cohen004] Florent Cohen, Philippe Decaudin, and Fabrice Neyret. Gpu-based lighting and shadowing of complex natural scenes. In SIGGRAPH '04: ACM SIGGRAPH 2004 Posters, page 91, New York, NY, USA, 2004. ACM.

[CG_Debev998] P. Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In SIGGRAPH '98: Proc. of the 25th annual conference on Computer graphics and interactive techniques, 189-198, 1998.

[CG_Edelsbr94] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. ACM Transactions on Graphics, 13(1):43–72, 1994

- [CG_Felke004] Felkel, P., Kanitsar, A., Fuhrmann, A., Wegenkittl, R.: Surface Models of Tube Trees. In: Computer Graphics International'04. (2004) 70–77
- [CG_Figue996] de Figueiredo, L.H., Gomes, J.: Sampling Implicit Objects with Physically-Based Particle Systems. In: Computer Graphics, Vol. 20. (1996) 365–376326
- [CG_Gilet005] Gilet G, Meyer A, Neyret F. Point-based rendering of trees. In Eurographics Workshop on Natural Phenomena, Poulin P, Galin E (eds). ACM/SIGGRAPH: Dublin, Irlande 2005; 67–72.
- [CG_Jaeger010] Marc Jaeger, Jinyuan Jia, Xiaopeng Zhang, Sebastien Griffon. VR Web based tree and vegetation representations for environmental applications and studies, an example on ChongMing Island. LandMod 2010, Montpellier, February 3-5, 2010, France (to appear)
- [CG_JT007a] Jun Teng, Baogang Hu, Marc Jaeger. Fast Tree Ambient Occlusion Approximation. In: T. Fourcaud and XP. Zhang, (Eds), Proceedings of Plant growth Modeling, simulation, visualization and their Applications, IEEE Computer Society (Los Alamitos, California), 2007, p. 319-322
- [CG_JT007b] Jun Teng, Marc Jaeger and Baogang Hu. A Fast Ambient Occlusion Method for Real-Time Plant Rendering. Journal of Computer Sciences and Technology vol. 22, no. 6, 2007, p. 859-866
- [CG_Kharla008] Kharlamov A., Cantlay I, and Stepanenko Y., 2008. Chapter 4. Next-generation SpeedTree rendering, in Nguyen H. (Eds), GPU Gems 3. Canada, Addison-Wesley, p. 69-92
- [CG_Lang000] Langer, M.S.; H. H. Buelthoff (2000). "Depth discrimination from shading under diffuse lighting". Perception 29 (6): 649-660.
- [CG_Meyer001] Meyer A, Neyret F, Poulin P. Interactive rendering of trees with shading and shadows. In Proceedings of 2001 Eurographics Workshop on Rendering Techniques, 2001, 183–196.
- [CG_Bechmann007] D. Bechmann , B. Péroche. 2007. Informatique graphique, modélisation géométrique et animation. Hermes Science Traité IC2, 2007, 372 pages ISBN 978-2-7462-1514-6
- [CG_Reeves985] William T. Reeves and Ricki Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. SIGGRAPH Comput. Graph., 19(3):313–322, 1985.
- [CG_Reeves987] Reeves WT, Salesin D, Cook RL. Rendering antialiased shadows with depth maps. Computer Graphics (Proceedings of SIGGRAPH'87), 1987, 283–291.
- [CG_Ricci073] A. Ricci. A constructive geometry for computer graphics. The Computer Journal, 16:157–160, May 1973.
- [CG_Sclarof991] S. Sclaroff and A. Pentland. Generalized implicit functions for computer graphics. In Computer Graphics, volume 25, pages 247–250. ACM Press, July 1991.
- [CG_Sousa008] Sousa T., 2008. Chapter 16. Vegetation Procedural Animation and Shading in Crysis, in Nguyen H. (Eds), GPU Gems 3. Canada, Addison-Wesley, p. 105-121
- [CG_Witki094] Witkin, A.P., Heckbert, P.S.: Using Particles to Sample and Control Implicit Surfaces. In: Proc. SIGGRAPH'94. (1994) 269-277
- [CG_XM006] Xing Mei, Marc Jaeger, Baogang Hu. An Effective Stratified Sampling Scheme for Environment Maps with Median Cut Method. In: Springer Verlag Lecture Notes of Computer Sciences. Proceedings of CGIV06 3rd International Conference on Computer Graphics, Imaging and visualisation, Sydney, Australia, July 26-28, 2006. Univ. of Technology of Sydney, IEEE Computer Society, p. 384-389
- [CG_Yin005] WeiWei Yin, Marc Jaeger, Jun Teng, Baogang HU. Modelling and Sampling Ramified Objects with SubStructure based Method. V.S. Suderam et al. (Eds): ICCS 2005, Atlanta, May 2005 USA. LNCS 3515 p. 322-326, 2005. Springer-Verlag Berlin Heidelberg 2005

Modèle de plantes (par règles)

- [MP_Aono984] Aono 'M and Kunii T L 1984 Botanical tree image generation; IEEE Comput. Graphics Appl. 4 10-33

- [MP_Barthe007] Barthelemy, D. and Caraglio, Y., 2007. Plant architecture: a dynamic, multilevel and comprehensive approach to plant form, structure and ontogeny. In: *Annals of Botany*, Vol.1, No 33, pp. :1-33.
- [MP_Blaise998] Blaise F. , Barczy J.-F., Jaeger M., Dinouard P., de Reffye P., 1998. Simulation of the growth of plants – Modeling of metamorphosis and spatial interactions in the architecture and development of plants. In: *Cyberworlds*. Springer-Verlag, Tokyo, p. 81-109, 1998.
- [MP_Courne005] Cournède P. and de Reffye P., 2005. Dynamical Model of Plant Growth for Environmental Applications. *ERCIM News*, 61, avril 2005.
- [MP_Davie000] Daviero, V., Meyer-Berthaud, B., Lecoustre, R., 2000. Computer simulation of sphenopsids architecture. I.Principles and methodology. *Review of Palaeobotany and Palynology*, 109, 121-134.
- [MP_Dauza008] Dauzat, Jean; Clouvel, Pascal; Luquet, Delphine; Martin, Pierre, 2008. Using Virtual Plants to Analyse the Light-foraging Efficiency of a Low-density Cotton Crop. *Annals of Botany*, Volume 101, Number 8, 3 May 2008 , pp. 1153-1166(14)
- [MP_Deussen004] Deussen, O. and Lintermann, B. 2004 *Digital Design of Nature: Computer Generated Plants and Organics*. SpringerVerlag.
- [MP_Dinouard988] Dinouard P., De Reffye P., Jaeger M., 1988. "Modelisation and simulation of rubber tree architecture ". In : Jacob J.L. (ed.), Prévôt J.C. (ed.). - *Compte-rendu du colloque exploitation-physiologie et amélioration de l'hévéa*. [Proceedings of the colloquium on exploitation-physiology and improvment of hevea] Hevea 88; 1988/11/02-07; Paris , Montpellier (FRA) : CIRAD-IRCA, 1988/11. p. 513-517.
- [MP_Dinouard987] Dinouard P., De Reffye P., Jaeger M., 1987. "Modélisation et simulation de l'architecture de l'hévéa ". *Revue Générale des Caoutchoucs et Plastiques (FRA)* , 1987/11. vol. 64, no. 673, p. 53-55.
- [MP_Eyroll986] Eyrolles G, Francon J and Viennot G 1986 Combinatoire pour la synthèse d'images réalistes de plantes; in *Actes du deuxième colloque Image*, CESTA 648-652
- [MP_Godin000] Godin, C., 2000. Representing and encoding plant architecture: a review. *Annals of Forest Science*, 57(05-juin): 413-438.
- [MP_Halle978] Hallé F, Oldeman R A A and Tomlison P B 1978 *Tropical trees and forests—An architectural analysis* (Berlin: Springer)
- [MP_Homazi000] El Homazi M. A., Zirari A., Dollé V., Dauzat J., Jaeger M., Lecoustre R., Oihabi, A., 2000. Modeling of the architecture of the date palm. Methodology and applications. In : *Date palm international symposium*, Sam Nujoma (Ed.), Windhoek, Namibia, 22-25/03/2000, p. 45-46.
- [MP_Lecoustre992] Lecoustre, R., de Reffye, P., Jaeger, M., and Dinouard, P. 1992. Controlling the architectural geometry of a plant's growth—application to the Begonia genus. In *Creating and Animating the Virtual World*, N. Magnenat Thalmann and D. Thalmann, Eds. Springer Computer Animation Series. Springer-Verlag New York, New York, NY, 199-214.
- [MP_Li010] Li, Z and Le Chevalier, V. and Courbède, P.H., 2010, Towards a Continuous Approach of Functional-Structural Plant Growth, *Proceedings of Plant Modeling and Applications (PMA09)*. Beijing, IEEE, CPS, pp. 334-340
- [MP_Linden968] A.Lindenmayer. Mathematical models for cellular interaction in development. *Journal of Theoretical Biology*, 18:280–315, 1968.
- [MP_Jaeger985] Jaeger M., 1985. Représentation de végétaux. (Master report, in French) Mémoire (DEA Traitement graphique, Traitements d'images), Strasbourg (FRA) : Université LouisPasteur, 1985. - 62 p.
- [MP_Jaeger987] Jaeger M., 1987. "Représentation et simulation de croissance des végétaux ". (PhD dissertation, in French) Thèse (Doctorat d'Université en Informatique), Strasbourg (FRA). Université Louis Pasteur, 1987. - 156 p.
- [MP_Jaeger992] Jaeger M., De Reffye P., 1992. "Basic concepts of computer simulation of plant growth". In : *Journal of Biosciences*, 1992. vol. 17, no. 3, p. 275-291.
- [MP_Jaeger010] Marc Jaeger, Ruoxi Sun, Jinyuan Jia, Vincent Le Chevalier. Efficient virtual plant data

structure for visualization and animation. In: Proceedings of IADIS International Conferences Computer Graphics, Visualization, Computer Vision and Image Processing 2010, MCCSIS 2010 Freiburg, Germany, July 27-29, ISBN 978-972-8939-22-9, 2010 IADIS, pp. 65-75

[MP_Oldema974] Oldeman R A A 1974 L'architecture de la foret guyanaise (Paris: Orstom)

[MP_Oppen986] Oppenheimer P E 1986 Real time design and animation of fractal plants and trees (Proc. SIGGRAPH

[MP_Prunsin988] Prusinkiewicz, P., Lindenmayer, A., and Hanan, J. 1988. Developmental models of herbaceous plants for computer imagery purposes. Computer Graphics 22, 4, 141–150.

[MP_Prunsin990] P. Prusinkiewicz and A. Lindenmayer. The Algorithmic Beauty of Plants. Springer Verlag, New York, NY, USA, 1990.

[MP_deReffye985] De Reffye P., Edelin C., Jaeger M., Cabart C. Simulation de l'architecture des arbres. In: L'arbre, Colloque International de l'Arbre, 1985, Montpellier, France Institut de Botanique, 1986. 17 p.

[MP_deReffye988c] De Reffye P., Edelin C., Jaeger M., Blaise F., Fournie G., Bree Lefebvre V., 1987. "L'arbre et l'ordinateur ". In : Pleasure of landscape , IFLA World Congress. 14; 1987/09/01-04; Paris, France IFLA, 1988. - p.198-201.

[MP_deReffye988b] De Reffye P., Cognée M., Jaeger M., Traoré B., 1988. "Modélisation stochastique de la croissance et de l'architecture du cotonnier. 1. Tiges principales et branches fructifères primaires ". Coton et Fibres Tropicales, France , 1988. vol. 43, no. 4, p. 269-291.

[MP_deReffye988a] De Reffye P., Jaeger M., Edelin C., Françon J., Puech C., 1988. " Plant models faithful to botanical structure and development". In: Computer Graphics , Siggraph 1988. vol. 22, p. 151-158.

[MP_deReffye989] De Reffye P., Edelin C., Jaeger M., 1989. "La modélisation de la croissance des plantes ". La Recherche (FRA), 1989. vol. 20, no. 207, p. 158-168.

[MP_deReffye003a] De Reffye, P. and Hu, B., 2003. Relevant and qualitative and quantitative choices in building an efficient dynamic Plant growth model: GreenLab case. Proc. of Plant Modeling and Applications (PMA03). Beijing, Springer Tsinghua, 2003. pp. 87-107

[MP_deReffye003b] De Reffye, P. and Goursat, M. and Quadrat J.P. and Hu, B., 2003. The dynamic equations of the tree morphogenesis GreenLab model. Proc. of Plant Modeling and Applications (PMA03). Beijing, Springer Tsinghua, 2003. pp. 108-117

[MP_deReffye008b] de Reffye P., Jaeger M., Cournède P.H, 2008. Le monde végétal. In: Comité International des Jeux Mathématiques (Ed) Maths Nature Express. Paris, CILM, p. 26-35

[MP_deReffye008a] P. de Reffye, D. Barthélémy, P.-H. Cournède, and M. Jaeger. Modélisation et simulation de l'architecture et de la production végétales. In F. Hallé, editor, Aux origines des plantes. Fayard, 2008

[MP_Sun009] Ruoxi Sun, Jinyuan Jia and Marc Jaeger, "Intelligent Tree Modeling Based on L-system". IEEE 10th International Conference on Computer-aided Industrial Design & Conceptual Design. November 26-29, 2009, Wenzhou, China

[MP_Smith984] Smith A R 1984 Plants, fractals and formal languages; Comput. Graphics 18 1-10

[MP_Weber995] Weber J., and Penn, J. 1995. Creation and rendering of realistic trees. In Proceedings of SIGGRAPH 1995, 119–128.

[MP_Yan003] Yan H., Jean François B., De Reffye P., Hu B., Jaeger M., Leroux J. 2003. Fast Algorithms of Plant Computation Based on Substructure Instances. International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision, vol. 3, no. 10, 2003, p. 145-153 [MP_Kawagu982] Kawaguchi Y 1982 A morphological study of the form of nature (Proc. SIGGRAPH '82); Comput. Graphics 16 223-232

[MP_Zhao003] Zhao, X. and de Reffye, P. and Barthelemy, D. and Hu, B., 2003. Interactive simulation of plant architecture based on a dual-scale automaton model. Proceedings of Plant Modeling and Applications (PMA03). Beijing, Springer Tsinghua, 2003. pp. 144-153.

Reconstruction de plantes à partir d'images ou points laser

- [MRP_Cheng007] Cheng, C.; Zhang, X. & Chen, B. (2007). Simple reconstruction of tree branches from a single range image. *Journal of Computer Science and Technology* , Vol.22, No.6, pp. 846-858, ISSN: 1000-9000 (print version) ISSN: 1860-4749 (electronic version).
- [MRP_Neuber007] Neubert, B., Franken, T., AND Deussen, O. 2007. Approximate image-based tree-modeling using particle flows. In *Proceedings of SIGGRAPH 2007*.
- [MRP_Ning010] Ning, X., Xhang, X., and Wang, Y. 2009. Tree segmentation from scanned scenes data. In *Proceedings of PMA09.*, IEEE Press.
- [MRP_Ning009] Xiaojuan Ning, Xiaopeng Zhang, Yinghui Wang, Marc Jaeger. 2009. Segmentation of architecture shape information from 3D point cloud. In *Proceedings of the 8th international Conference on Virtual Reality Continuum and Its Applications in industry (Yokohama, Japan, December 14 - 15, 2009)*. S. N. Spencer, Ed. VRCAI '09. ACM, New York, NY, pp. 127-132
- [MRP_Quan006] L. Quan, P. Tan, G. Zeng, L. Yuan, J. Wang, and S. B. Kang. Image-based plant modeling. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 599–604, 2006. Also published in: *ACM Transactions on Graphics* 25, 3 (Aug.), 772–778.
- [MRP_Reche004] A. Reche-Martinez, I. Martin, and G. Drettakis. Volumetric reconstruction and interactive rendering of trees from photographs. *ACM Transactions on Graphics*, 23(3):720–727, 2004.
- [MRP_Sun009] Ruoxi Sun, Jinyuan Jia, Hongyu Li, Marc Jaeger. Image-based Lightweight Tree Modeling. In: *Proceedings of the 8th international Conference on Virtual Reality Continuum and Its Applications in industry (Yokohama, Japan, December 14 - 15, 2009)*. S. N. Spencer, Ed. VRCAI '09. ACM, New York, NY, pp. 17-22
- [MRP_ShlyakT001] I. Shlyakhter, M. Rozenoer, J. Dorsey, and S. Teller. Reconstructing 3d tree models from instrumented photographs. *IEEE Computer Graphics and Applications*, 21(3):53–61, 2001. 1
- [MRP_Tan007] Tan, P., Zeng, G., Wang, J. D., Kang, S. B., and Quan, L. 2007. Image-based tree modeling. In *Proceedings of SIGGRAPH 2007*.
- [MRP_Tan008] Tan P., Fang T., Xiao J.X., Zhao P., Quan L., 2008. Single image tree modeling. *ACM Transactions on Graphics*. 27(5): 108
- [MRP_Xu007] Xu, H.; Gossett, N. & Chen, B. (2007). Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Transactions on Graphics* , Vol. 26, No.4, pp. 19. ISSN: 0730-0301.
- [MRP_Zhu008] Chao Zhu, Xiaopeng Zhang, Baogang Hu, Marc Jaeger. Reconstruction of Tree Crown Shape from Scanned Data. In: *Proceedings of Third International Conference of E-Learning and Games - Edutainment 08*. Nanjing, China. Z. Pan et al. (Eds.): Edutainment 2008, LNCS 5093, p. 745-756, 2008.
- [MRP_Zhu010] Chao Zhu, Xiaopeng Zhang and Marc Jaeger. New Method for Construction of Tree Crown from Scanned Data. In: Li, B. and Jaeger, M. and Guo, Y. (Eds). 2010. *Proceedings of Plant growth Modeling, and their Applications (PMA09)*, Beijing, China, November 9-13, 2009, IEEE CPS, pp. 352-359

Modèle de Plante construits interactivement (Sketch based)

- [MPU_Bloom985] Bloomenthal J. Modeling the mighty maple. *SIGGRAPH Computer Graphics* 1985; 19(3): 305–311.
- [MPU_Chen008] Chen, X. J., Neubert, B., Xu, Y. Q., Deussen, O., and Kang, S. B. 2008. Sketch-based tree modeling using markov random field. In *ACM SIGGRAPH Asia*.
- [MPU_Deussen996] Deussen O, Lintermann B. Interactive modelling of branching structures. In *Proceedings of the ternational Conference on Computer Graphics and Interactive Techniques*, 1996, 130–135.

- [MPU_Deussen997] Deussen, O. and Lintermann, B. 1997. A modelling method and user interface for creating plants. *Proceedings of Graphics Interface 97*, Morgan Kaufmann Publishers, pp. 189-197
- [MPU_Gardne984] Gardner G 1984 Simulation of natural scenes using textured quadric patches; *Comput. Graphics* 18 11-20
- [MPU_Linter999] Lintermann, B. and Deussen, O. 1999. Interactive structural and geometrical modeling of plants. *IEEE Computer Graphics and Applications*, 19, 1 pp. 56-65
- [MPU_Wither009] Wither, J., Boudon, F., Cani, M. P., and Godin, C. 2009. Structure from silhouettes: a new paradigm for fast sketch-based design of trees. In *Proceedings of EUROGRAPHICS 2009*.

Modelisation de systèmes complexes. Modélisation de paysages, forêts et de systèmes cultureux

- [MSL_Alsweis005] M. Alsweis, O. Deussen, 2005. Modeling and Visualization of Symmetric and Asymmetric Plant Competition. In: *Eurographics Workshop on Natural Phenomena*, Dublin, 2005 August 30th, pages 83-88, 2005
- [MSL_Campos998] Campos A., Hill D., "Web-based Simulation of Agent Behaviors", *SCS International Conference on Web-based Modeling and Simulation* ", 11-14 Janvier, San Diego, USA, pp. 9-14, 1998
- [MSL_Coquillard997] Coquillard P., Hill D. 1997. Modélisation et Simulation des Ecosystèmes. Des modèles déterministes aux simulations à événements discrets. 1997, Masson, 273 p.
- [MSL_Coligny007] De Coligny F. , 2007. Efficient Building of Forestry modelling software with the Capsis methodology. In: Fourcaud T, Zhang XP, eds. *Plant Growth Modeling and Applications. Proceedings of PMA06*. Los Alamitos, California: IEEE Computer Society, pp. 216-222.
- [MSL_Cournede010] Paul-Henry Cournède, Thomas Guyard, Benoît Bayol, Sébastien Griffon, François de Coligny, Philippe Borianne, Marc Jaeger and P. de Reffye. A Forest Growth Simulator Based on Functional-Structural Modelling of Individual Trees. In: Li, B. and Jaeger, M. and Guo, Y. (Eds). 2010. *Proc. of Plant growth Modeling, and their Applications*, Beijing, China, November 9-13, 2009, IEEE CPS, pp. 34-41
- [MSL_Deussen998] Oliver Deussen and Pat Hanrahan and Bernd Lintermann and Radom Mech and Matt Pharr and Przemyslaw Prusinkiewicz, 1998. Realistic Modeling and Rendering of Plant Ecosystems. In *SIGGRAPH 98 Conference Proceedings*, Computer Graphics, vol. 32, pages 275-286, July 1998
- [MSL_Edelin989] Edelin, C., de Reffye, P., Jaeger, M., Dinouard, P., 1989. La simulation de l'architecture des arbres et son rôle potentiel dans la conception et la gestion des paysages urbains. *Revue Forestière Française*, No 41 (n sp. "L"), pp. 143-153.
- [MSL_Gaucherel006] Gaucherel C., Giboire N., Viaud V., Houet T., Baudry J. and Burel F., 2006. A domain specific language for patchy landscape modelling: the brittany agricultural mosaic as a case study. *Ecological Modelling*. 194, p. 233-243.
- [MSL_Griffon009] Griffon S., Auclair D. (2009). Visualising changes in agricultural landscape. In Van Ittersum M.K., Wolf J., Van Laar H.H. (Eds). *Proceedings of AgSAP*, Egmond aan Zee, The Netherlands. Wageningen University and Research Centre, 10-12 March 2009, p. 296-297.
- [MSL_Green989] N. Green. 1989. Voxel space automata: Modeling with stochastic growth processes in voxel space. In *Proc. SIGGRAPH'89*, New York, USA, 1989, p. 175-184
- [MSL_Hammes001] J. Hammes. Modeling of Ecosystems as a Data Source for Real-Time Terrain Rendering. *LNCS 2181*, p. 98. *Digital Earth Moving First International Symposium, DEM 2001*, Manno, Switzerland, September 5-7, 2001, Springer Verlag
- [MSL_Hill994] Hill D., Pastre J., Coquillard P., Gueugnot J., "Design of an Ecosystem Modelling Environment : Application to Forest Growth Simulation", *CISS 94, Joint Simulation Societies*, Zurich, Switzerland, August 22-26 1994, pp. 538-544
- [MSL_Hill009] Hill D., "Simulation Informatique au Service des Sciences de la Vie", *Université Blaise Pascal*, 2009, 210 p.

- [MSL_LandMod010] 2010 International Conference on Integrative Landscape Modelling. Montpellier, France February 3-5, 2010. An initiative of: <http://www.umr-lisah.fr/rtra-projects/projects.html>
- [MSL_leChevalier007a] Vincent Le Chevalier, Aurélien Lesluye, Marc Jaeger, Xing Mei, and Paul-Henry Cournède. A Functional Landscape Prototype to simulate Water Resource competition between Plants. In : T. Fourcaud and XP. Zhang, (Eds), Proceedings of Plant growth Modeling, simulation, visualization and their Applications, IEEE Computer Society (Los Alamitos, California), 2007, pp. 124-131
- [MSL_leChevalier007b] Vincent Le Chevalier, Marc Jaeger, Xing Mei, Paul-Henry Cournède. Simulation and Visualisation of Functional Landscapes: Effects of the Water Resource Competition between Plants. Journal of Computer Sciences and Technology. vol. 22, no. 6, 2007, p. 835-845
- [MSL_leChevalier010b] Vincent Le Chevalier, Marc Jaeger. Bottom-up approach of landscape simulation leading to a generic synchronization. In Proceedings of LandMod 2010, Montpellier, February 3-5, France
- [MSL_leChevalier010a] Vincent Le Chevalier, Marc Jaeger and Paul-Henry Cournède. Synchronisation Formalism, Resource and Plant Models for Plant Ecosystem Simulation. In: Li, B. and Jaeger, M. and Guo, Y. (Eds). 2010. Proceedings of Plant growth Modeling, and their Applications (PMA09), Beijing, China, November 9-13, 2009, IEEE CPS, pp. 277-284
- [MSL_Neidhold005] B. Neidhold, M. Wacker, and O. Deussen, 2005. Interactive physically based Fluid and Erosion Simulation. In: Eurographics Workshop on Natural Phenomena, Dublin, 2005 August 30th, p. 25-32
- [MSL_Quesnel993] G. Quesnel, R. Duboz and E. Ramat. The Virtual Laboratory Environment - An Operational Framework for Multi-Modelling, Simulation and Analysis of Complex Systems, Simulation Modelling Practice and Theory, (17), 641-653, April 2009.
- [MSL_Touraille010] Touraille L., Traore M.K. and Hill D., SimStudio : une Infrastructure pour la Modélisation, la Simulation et l'Analyse de Systèmes Dynamiques Complexes, UMR CNRS 6158, LIMOS , RR-10-13, 2010, 12 p.
- [MSL_Ziegler000] B. P. Zeigler, H. Praehofer, and T. G. Kim, Theory of Modeling and Simulation. Integrating Discrete Event and Continuous Complex Dynamic Systems. Second edition, Academic Press, Jan. 2000
- [MSL_Ziegler993] Bernard P. Zeigler , Sankait Vahie, DEVS formalism and methodology: unity of conception/diversity of application, Proceedings of the 25th conference on Winter simulation, p.573-579, December 12-15, 1993, Los Angeles, California, United States

Modèles multi-résolution et rendu à niveaux de détails

- [LOD_Deng007a] Qingqiong Deng, Xiaopeng Zhang, Xiangdong Lei, Marc Jaeger. Fast Forest Visualization on Hierarchical Images and Visibility, Proceedings of FSPM2007, the 5th International Workshop on Functional Structural Plant Models, November 4-9, 2007, Napier, New Zealand.
- [LOD_Deng007b] Qingqiong Deng, Xiaopeng Zhang, Marc Jaeger. View-Dependent Hierarchical Foliage Simplification. In: Technologies for E-learning and Digital Entertainment: Second International Conference, Edutainment 2007, Hong Kong, China, June 11-13, 2007, K.-C. Hui, Z.-G. Pan, R.-C.-K. Chung, C.-C.-L. Wang, X.-G. Jin, S. Gobel, C.-L. Li (editors), LNCS, Springer, p. 44-55, June 2007.
- [LOD_Deng007c] Qingqiong Deng, Xiaopeng Zhang, Sebastien Gay, and Xiangdong Lei. Continuous lod model of coniferous foliage. International Journal of Virtual Reality, 6(4):77-84, 2007.
- [LOD_Deng008] Qingqiong Deng and Xiaopeng Zhang. Grid-based view-dependent foliage simplification. Journal of Computational Information Systems, 4(4):1643-1650, 2008.
- [LOD_Deng010] QingQiong Deng, Xiaopeng Zhang, Gang Yang, Marc Jaeger. Multiresolution foliage for forest rendering, in Computer Animation and Virtual Worlds, 2010, Vol. 21, 1, John Wiley and Sons, pp. 1-23
- [LOD_Fuhrma005] Fuhrmann A, Umlauf E, Mantler S. Extreme model simplification for forest rendering. In Proceedings of 2005 Eurographics Workshop on Natural Phenomena, 2005, 57-66.

- [LOD_Hoppe996] H. Hoppe. Progressive meshes. ACM SIGGRAPH 1996 Conference Proceedings, 99-108.
- [LOD_Hoppe998] H. Hoppe. Smooth view-dependent level-of-detail control and its application to terrain rendering. IEEE Visualization 1998 Conference, p. 35-42.
- [LOD_Melax998] Melax S. A simple, fast and effective polygon reduction algorithm. Game Developer magazine; 1998, p. 44-9 {<http://www.melax.com/polychop/gdmag.pdf>}
- [LOD_Li010] Hongjun Li, Xiaopeng Zhang, Wujun Che and Marc Jaeger. Smooth Transition between Different Plant Leaves Models. In: Li, B. and Jaeger, M. and Guo, Y. (Eds). 2010. Proc. of Plant growth Modeling, and their Applications (PMA09), Beijing, China, November 9-13, 2009, IEEE CPS, pp. 376-383
- [LOD_Remolar002] Remolar I, Chover M, Belmonte O, Ribelles J. Geometric simplification of foliage. In Proceedings of Eurographics 2002 Conference-Short Presentations, 2002, 397-404.
- [LOD_Remolar003] Remolar I, Chover M, Ribelles J, Belmonte O. View-dependent multiresolution model for foliage. In Proceedings 11-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, 2003, 370-378.
- [LOD_Rebollo006] Rebollo C, Remolar I, Chover M, Gumbau J. Hardware-oriented visualisation of trees. In Proceedings of the 21st International Symposium on Computer and Information Sciences, 2006, 374-383.
- [LOD_Zhang003] Zhang X, Blaise F. Progressive polygon foliage simplification. In Proceedings of the International Symposium on Plant growth Modeling and their Applications, 2003, pp. 217-230.
- [LOD_Zhang006b] Xiaopeng ZHANG, Qingqiong DENG, Marc JAEGER. Level of detail technique for plant models. Chinese Journal Of Stereology and Image Analysis, 2006, vol.11, no.4, p. 235-242
- [LOD_Zhang006a] Xiaopeng Zhang, Frederic Blaise, Marc Jaeger. Multiresolution Plant Models with complex organs. Proceedings of ACM VRCIA 2006, p. 331-334, June 14-17 th Hong-Kong, China

Rendu realiste de forets par images (billboards)

- [RRB_Bao010] Guanbo Bao, Xiaopeng Zhang, Wujun Che and Marc Jaeger. Billboards for Tree Simplification and Real-time Forest Rendering. In: Li, B. and Jaeger, M. and Guo, Y. (Eds). 2010. Proc. of Plant growth Modeling, and their Applications (PMA09), Beijing, China, November 9-13, 2009, IEEE CPS, pp. 433-440
- [RRB_Behren005] S. Behrendt, C. Colditz, O. Franzke, J. Kopf, and O. Deussen. Realistic realtime rendering of landscapes using billboard clouds. In Proceedings of Eurographics 2005, pages 507-516, 2005.
- [RRB_Candus005] A. Candussi, N. Candussi, and T. Hollerer. Rendering realistic trees and forests in real time. In Proc. Eurographics 2005, pages 73-76, August 2005.
- [RRB_Deussen002] Deussen O, Colditz C, Stamminger M, Drettakis G. Interactive visualization of complex plant ecosystems. In Proceedings of the IEEE Visualization Conference, 2002, 219-226.
- [RRB_Decore003] Xavier Decoret, Frédo Durand, François X. Sillion, and Julie Dorsey. Billboard clouds for extreme model simplification. In SIGGRAPH '03: ACM SIGGRAPH 2003 Papers, pages 689-696, New York, NY, USA, 2003. ACM.
- [RRB_Fernan007b] Fernández, I.G. 2007. Generation and Interactive Visualization of 3D vegetation. Master Thesis, Master in Computing. http://www.gametools.org/archives/publications/mastertesi_isma.pdf Universitat de Girona, 2007.
- [RRB_Fernan007a] Ismael Garcia Fernandez, Gustavo Patow, Mateu Sbert, and Laszlo Szirmay-Kalos. Multi-layered indirect texturing for tree rendering. In S. M W rillou D. Ebert, editor, Eurographics Workshop on Natural Phenomena, 2007.
- [RRB_Fuhrma005] A. Fuhrmann, E. Umlauf, and S. Mantler. Extreme model simplification for forest rendering. In Proc. of 2005 Eurographics Workshop on Natural Phenomena, pages 57-66, 2005.
- [RRB_Honjo001] Honjo, T. and Lim, E., 2001. Visualization of landscape by VRML system, Landscape and

Urban Planning, 55, 175-183.

[RRB_Honjo010] Honjo T., Umeki K., Lim E., Wang D.H. Yang P.A., Hsieh H.C., 2009. Landscape Visualization on Google Earth. Accepted to PMA09, 9-12 November 2009, Beijing, China, to appear [RRB_Jakuli000] Jakulin A. Interactive vegetation rendering with slicing and blending. In Proceedings of the Eurographics 2000 Conference-Short Presentations, 2000, 273-282.

[RRB_Max996] Max N. Hierarchical rendering of trees from pre-computed multi-layer z-buffers. In Proceedings of the 1996 Eurographics Workshop on Rendering, 1996, 165-174.

[RRB_Max999] Max N, Deussen O, Keating B. Hierarchical image-based rendering using texture mapping hardware. In Proceedings of the 1999 Eurographics Workshop on Rendering, 1999, 57-62.

Imagerie Volumique

[IV_Banegas001] Banégas F., Jaeger M., Michelucci D., Roelens M., 2001. The ellipsoidal skeleton in medical applications. In : 6th ACM Symposium on Solid modeling and Applications, Ann Arbor, Michigan USA 4-8/06/2001, ACM

[IV_Banegas999a] Banégas F., Michelucci D., Roelens M., Jaeger M., Canovas F., 1999. Hierarchical automated clustering of cloud point set by ellipsoidal skeleton : application to organ geometric modeling from CT-scan images. Session 3661-128. SPIE's International Symposium in Medical Imaging 1999, 20-26 February 1999, San Diego, USA

[IV_Briand993b] Briand D., Jaeger M., Kiriakopoulou T., Bonnel F., Joyeux H., 1993. 3D reconstruction and computing volume of tumorous liver. Therapeutic interest of repeat CT scans during chemotherapy for hepatic metastases. In: 2nd European Congress of the European Association of Clinical Anatomy, Munich (Germany), 5-7 sept. 1993, p. 219.

[IV_Briand993a] Briand D., Jaeger M., Borianne Ph., Bonnel F., 1993. Reconstruction anatomique tridimensionnelle. Visualisation par lancer de rayon à partir de coupes tomодensitométriques. In: 75ème Congrès de l'Association des Anatomistes de France, Lille (France), 12-14 mai 1993.

[IV_Bonnel000] Bonnel F., Borianne P., Bonnel C., Jaeger M., Cyteval C., 2000. Biometrics Three-dimensional of Fore-foot assisted by computer C2000. In : AFCP, 2nd international spring meeting, Bordeaux, France, May 4-5-6 2000

[IV_Canovas000] Canovas, F., Banegas, F., Cyteval, C., Jaeger, M., DiMeglio, A., Bonnel, F., & Sultan, C. 2000. Carpal bone maturation assessment by image analysis from computed tomography scans. Hormone Research. vol. 54, no. 1, p. 6-13.

[IV_Chemouny999] Chemouny S., Joyeux H., Borne F., Jaeger M., Monga O., 1999. Advanced 3D image processing techniques for liver and hepatic tumors location and volumetry. SPIE's International Symposium on Medical Imaging 1999, vol. 3661-I, p. 761-771, 20-26 february 1999, San Diego, USA.

[IV_Cline988] Cline H.E. & al., 1988. Two algorithms for the three-dimensional reconstruction of tomograms. In: Medical Physics, Vol. 15, No 3, May/June 1988, pp. 320-327.

[IV_Cristol996] Cristol-Gaubert R., Prudhomme M., Jaeger M., de Reffye P., Godlewski G., 1996. Reconstruction tridimensionnelle de l'arbre biliaire à la fin de la période embryonnaire chez l'homme et chez le rat. 78e congrès de l'association des anatomistes de langue française. Madrid, le 27 septembre 1996.

[IV_Decaud004] P. Decaudin and F. Neyret. Rendering forest scenes in real-time. In Proc. of the 2004 Eurographics Workshop on Rendering, pages 93-102, 2004.

[IV_Decaud009] P. Decaudin and F. Neyret. Volumetric billboards. Computer Graphics Forum, 28(8):2079-2089, 2009.

[IV_Jaeger003] Jaeger M., Teng J., "Tree and Plant volume imaging. An introductive study towards voxelized fonctionnal landscapes". Proceedings of PMA03, Beijing October 2003. p. 169-181.

[IV_Jaeger999] Jaeger M., Leban J. M., Chemouny S., Saint André L., 1999. 3D stem reconstruction from

CT scan exams. .Third Workshop IUFRO WP S5.01-04, Topic 6, p. 399-409. Biological improvement of wood properties, La Londe-Les-Maures, France, 5-12 septembre 1999.

[IV_Jaeger993] Jaeger M., Briand D., Borianne Ph., Bonnel F., 1993. Knee anatomy 3D reconstruction and visualization from CT scans. In: 2nd European Congress of the European Association of Clinical Anatomy, Munich (Germany), 5-7 sept. 1993, p. 231.

[IV_Jin007] Changzhu Jin, Russell L. Ciochon, Wei Dong, Robert M. Hunt, Jr, Jinyi Liu, Marc Jaeger, and Qizhi Zhu. 2007. The first skull of the earliest giant panda. PNAS, June 26, 2007, vol. 104, no. 26, p. 10932-10937

[IV_Kaufm990] Kaufman A. (Ed.), Volume Visualization. IEEE Computer Society Press, 1990.

[IV_Lorens987] W.E. Lorensen and H.E. Cline. Marching cubes: a high resolution 3d surface algorithm. In Computer Graphics, volume 21, pages 163–169. ACM Press, july 1987.

[IV_Nicolini992] Nicolini, E., Guédon, Y., Jaeger, M., de Reffye, P., Escoute, J., Schwendiman, J., 1992. Modélisation et simulation de la croissance de *Agaricus bisporus* (Lange) Imbach. Bibliographie histocytologique. (Etude préliminaire) (Rapport Confidentiel). Montpellier, France : CACS & CIRAD - GERDAT Unité de Modélisation des Plantes.

[IV_Pfiste001] Pfister, H., Lorensen, B., Bajaj, C., Kindlmann, G., Schroeder, W., Avila, L. S., Martin, K., Machiraju, R., and Lee, J. 2001. The Transfer Function Bake-Off. IEEE Comput. Graph. Appl. 21, 3 (May. 2001), 16-22.

[IV_Treil994] Treil J., Casteigt J., Roch P., Jaeger M., Cavezian R., Pasquet G., 1994. La charpente maxillo-mandibulaire. Nouvelle approche crano-faciométrique tridimensionnelle. In: Actualités Odonto-Stomatologiques, n°188, déc. 1994, p. 627- 637.

[IV_Treil997] Treil J., Casteigt J., Madrid C., Jaeger M., Borianne P., 1997. Une nouvelle construction céphalométrique tridimensionnelle. Un nouveau paramètre d'analyse tridimensionnelle : les axes d'inertie. Un nouveau concept : l'équilibre maxillo-facial. In : Bulletin de l'Orthodontie française, 1997. Tome I, no 68, p. 171-181.

[IV_Wu010] Haiwen Wu, Marc Jaeger, Mao Wang, Baoguo Li and Baogui Zhang. 3D-Reconstruction and Visualization of Xylem Vessels of Wheat Nodal Root. In: Li, B. and Jaeger, M. and Guo, Y. (Eds). 2010. Proc. of Plant growth Modeling, and their Applications (PMA09), Beijing, China, November 9-13, 2009, IEEE CPS, pp. 384-390

Outils logiciels, normes

[ST_Bionatics010] Bionatics SA and LandSim 3D tool. 2010. <http://www.bionatics.com/>

[ST_Forester010] Forester tool. 2010 URL: <http://www.dartnall.f9.co.uk/forester/>

[ST_Geoma010] Geomantics Genesys 4 tool. 2010. URL: <http://www.geomantics.com/genesis4.htm>

[ST_Jaeger991] Jaeger, M., de Reffye, P., 1991. Le logiciel AMAP: un outil de simulation et de représentation des végétaux. In de Reffye, P., Jaeger, M. (Eds). L'informatique scientifique dans l'enseignement de la biologie et de la géologie au lycée : actes du colloque. Paris, France : Institut National de la Recherche Pédagogique, p. 243-248 (Colloque Technologies Nouvelles et Education).

[ST_Paper009] Tondeur P., Winder J., 2009. PaperVision3D Essentials, Packt Publishing, 428 p.

[ST_Qt010]. Qt cross-platform application and UI framework. <http://qt.nokia.com/products/>

[ST_SmartForest000] Smart Forest, 2000. Smart Forest, 2000. User's Manual, Version 6.5, 14 pp.

[ST_Simeo010] SIMEO. 2010. Free scene editor CAPSIS based. <http://amap-dev.cirad.fr/wiki/simeo/Simeo>

[ST_Terragen010] Terragen tool. 2010. URL : <http://www.planetside.co.uk/terrigen>

[ST_WCS010] World Construction Set, Visual Nature Studio, Scene Express. 2010. <http://3dnature.com/>

[ST_X3D007] Don Brutzman D., Leonard Daly L., 2007. X3D: Extensible 3D Graphics for Web Authors (The Morgan Kaufmann Series in Interactive 3D Technology). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 442 p.

Traitement d'image – Signal

[TIS_Banegas999] Banégas F., Michelucci D., Roelens M., Jaeger M., 1999. Automatic extraction of significant features from 3D point clouds by ellipsoidal skeletons. Applications in vision and geometric characterization. In : International Conference in Visual Computing 1999 (ICVC'99), S.P. Mudur, D. Shikhare, J.L. Encarnacao and J. Rossignac (Eds), Goa, Inde 23-26/02/1999. IFIP; p. 58-67.

[TIS_Borge986] G. Borgefors. Distance transformations in digital images. Computer Vision, Graphics and Image Processing, 34:344-371, 1986.

[TIS_Cai003] Cai ZhiFeng, Lu Hanqing, Jaeger Marc. Deformable Image Matching using Hybrid Elastic Models. Journal of Image and Graphics, vol. 8, no. 7, Jul 2003, p. 753-758

[TIS_Cai002] Zhifeng CAI, Hanqing LU, Marc Jaeger, "Non-rigid Image Registration Using Hybrid Elastic Models", Proceedings of ICDIA 2002, International Conference on Diagnostic Imaging and Analysis, Aug. 18-20, 2002, Shanghai, China. Shanghai Scientific and Technological Literature Publishing House, p. 145-150, 2002.

[TIS_Dai009] M. Dai, X. Zhang, Y. Zhang, M. Jaeger. Segmentation of Point Cloud Scanned from Trees, in "Proceedings of Workshop on 3D content and applications with ACCV 2009", September 24th, Xian, China, 2009 (to be published)

[TIS_Diday971] E. Diday. Une nouvelle méthode en classification automatique et reconnaissance des formes : la méthode des nuées dynamiques. In Rev. Statist. Appl., volume 19, pages 19–33, 1971.

[TIS_Derich987] Deriche R. 1987. Using Canny's criteria to derive a recursively implemented optimal edge detector. In: International Journal of Computer Vision, vol. 1, n°2, 1987, pp. 167-187.

[TIS_Nagao979] M. Nagao and T. Matsuyama. Edge preserving smoothing. Computer Graphics and Image Processing, 9:394–407, 1979.

[TIS_Zhang009] Xiaopeng Zhang, Jianfei Liu, Marc Jaeger and Zili Li. Volume Decomposition and Hierarchical Skeletonization. The International Journal of Virtual Reality, 2009, Vol 8, no. 1, p. 79-80

[TIS_Zhang010] Xiaopeng Zhang, Bo Xiang, Wujun Che, Marc Jaeger. Volume Decomposition and Hierarchical Skeletonization for Shape Analysis, Pattern Recognition, Peng-Yeng Yin (Ed.), ISBN: 978-953-307-014-8, INTECH, 2010

Annexe 1. Sélection de publications

Sélection choisie d'articles

Jaeger, M. and De Reffye, P. 1992. Basic concepts of computer simulation of plant growth. In: *Journal of Biosciences*. Springer India, in co-publication with Indian Academy of Sciences. Vol. 17, No3, p. 275-291. DOI=<http://dx.doi.org/10.1007/BF02703154/>

Jaeger, M., Leban, J.M., Borianne, P., Chemouny, S. and Saint-André, L., 1999. 3D stem reconstruction from CT scan exams. From log external shape to internal structures. In: *Connection between silviculture and wood quality through modelling approaches and simulation softwares*, 3rd Workshop IUFRO WP S5.01-04, G. Nepveu (Ed.), La Londe-Les-Maures, France 5-12/09/1999. INRA; pp. 399-109.

Banégas, F., Jaeger, M., Michelucci, D., and Roelens, M. 2001. The ellipsoidal skeleton in medical applications. In *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications* (Ann Arbor, Michigan, United States). D. C. Anderson, Ed. SMA '01. ACM, New York, NY, 30-38. DOI=<http://doi.acm.org/10.1145/376957.376961>

Vincent Le Chevalier, Marc Jaeger and Paul-Henry Cournède. Synchronisation Formalism, Resource and Plant Models for Plant Ecosystem Simulation. In: *Li, B. and Jaeger, M. and Guo, Y. (Eds). 2010. Proceedings of Plant growth Modeling, and their Applications (PMA09)*, Beijing, China, November 9-13, 2009, IEEE CPS, pp. 277-284

Marc Jaeger, Ruoxi Sun, Jinyuan Jia, Vincent Le Chevalier. Efficient virtual plant data structure for visualization and animation. In: *Proceedings of IADIS International Conferences Computer Graphics, Visualization, Computer Vision and Image Processing 2010*, MCCSIS 2010 Freiburg, Germany, July 27-29, ISBN 978-972-8939-22-9, 2010 IADIS, pp. 65-75

Sélection d'articles choisis téléchargeables librement (non inclus dans le dossier)

Jun Teng, Marc Jaeger and Baogang Hu. A Fast Ambient Occlusion Method for Real-Time Plant Rendering. In: *Journal of Computer Sciences and Technology* vol. 22, no. 6, 2007, p. 859-866
URL: <http://jcst.ict.ac.cn:8080/jcst/EN/article/downloadArticleFile.do?attachType=PDF&id=119>

Vincent Le Chevalier, Marc Jaeger, Xing Mei, Paul-Henry Cournède. Simulation and Visualisation of Functional Landscapes: Effects of the Water Resource Competition between Plants. In: *Journal of Computer Sciences and Technology*. vol. 22, no. 6, 2007, p. 835-845
URL: <http://jcst.ict.ac.cn:8080/jcst/EN/article/downloadArticleFile.do?attachType=PDF&id=118>

X.-P. Zhang, B. Xiang, W.J. Che and M. Jaeger. 2009. Volume Decomposition and Hierarchical Skeletonization for Shape Analysis, Pattern Recognition, Peng-Yeng Yin (Ed.), ISBN: 978-953-307-014-8, INTECH, Chapter 4, p.47-72 . URL: <http://sciyo.com/articles/show/title/volume-decomposition-and-hierarchical-skeletonization-for-shape-analysis>

Deng, Q., Zhang, X., Yang, G., and Jaeger, M. 2010. Multiresolution foliage for forest rendering. *Computer Animation and Virtual Worlds* 21, 1 (Jan. 2010), p. 1-23.
URL: <http://onlinelibrary.wiley.com/doi/10.1002/cav.283/pdf>

Marc Jaeger, Jinyuan Jia, Xiaopeng Zhang, Sebastien Griffon. VR Web based tree and vegetation representations for environmental applications and studies, an example on ChongMing Island. *Proceedings of LandMod 2010*, Montpellier, February 3-5, 2010, France. SympoScience publication, Quae Editions, 2010
URL: <http://www.symposcience.org/exl-php/articles/634-article.htm>

Basic concepts of computer simulation of plant growth

M JAEGER and PH DE REFFYE*

Centre de Cooperation Internationale en Recherche Agronomique pour le Developpement,
Departement GERDAT, Laboratoire de Modelisation, BP 5035, 34032 Montpellier Cedex 1,
France

MS received 14 December 1991; revised 13 May 1992

Abstract. The experimental study of plant growth and architecture requires two consecutive and complementary approaches. The first one is qualitative and botanical. Through this qualitative analysis, it is possible to divide a tree into botanically homogeneous subunits. The second one is quantitative. Modeling of the functioning of meristems based on stochastic processes has been carried out in the Laboratoire de Modelisation of CIRAD (1980 1990) in combination with extensive experimental work on temperate and tropical plants. Calculations involved in the tree simulations from field data are based on the probabilistic Monte Carlo method for the topological part and on analytical geometry for the morphological part. Data, such as length, angles, elasticities, tropism, are necessary for geometric construction of the plant. The growth model, together with the geometric calculations enables a random plant simulation in 3-D according to geometry for the morphological one. Computer graphics methods are then used to visualize the computed plant (perspective, colours, texture, shadows); they require special programs and graphic computers.

Key words. Plant growth; architecture; computer graphics; modeling; simulation; stochastic processes.

1. Introduction

Much progress has been made in understanding tree growth and architecture over the past 20 years. In 1970, a school specialized in morphological descriptions of tree architecture was founded by Hallé in Montpellier, a city known for its botanical studies for over 10 centuries. Hallé *et al* (1978), defined the basic concepts of architectural analysis, *i.e.* the architectural model and reiteration, which we will discuss further on. Botanists of this school developed a rational scientific method for visual representation, which is capable of expressing the main elements and growth strategies of trees.

In the field of computer-generated images, computer graphics of trees generated from algorithms already exist: specific branching processes (Kawaguchi 1982; Aono and Kunii 1984), graftals (Smith 1984), paracladial systems (Lindenmayer 1976; Prusinkiewicz and Hanan 1989), fractals (Oppenheimer 1986), combinatorial trees (Eyrolles *et al* 1986) and thin transparent ellipsoids (Gardner 1984).

Although these models are interesting from an algorithmic point of view as well as for their graphic results, they lack thorough botanical bases and an experimental approach. Except for the case of small plants which are obviously easier to model (Prusinkiewicz *et al* 1988), the trees created may only be seen from far, since details

*Corresponding author.

of the mathematical algorithm that produced them predominate when the trees are viewed at close quarters.

At CIRAD (Centre de Coopération Internationale en Recherche Agronomique pour le Développement), the Modeling Unit has formed a team of 20 researchers specialized in four different fields—botany, agronomy, mathematics and computer sciences—for quantitative approach to this problem. Our belief is that a well-balanced interaction of these disciplines is necessary to establish the fundamental principles of growth simulation and plant architecture (de Reffye *et al* 1988). In this paper we present the current status of our investigations.

2. Botanical basis of plant architecture

Halle has grouped all the known vascular plants into less than 30 architectural models according to simple morphological characteristics (samples are shown in figure 1). Each of these models corresponds to a particular combination of the different possible types of axes. This concept does not correspond to the

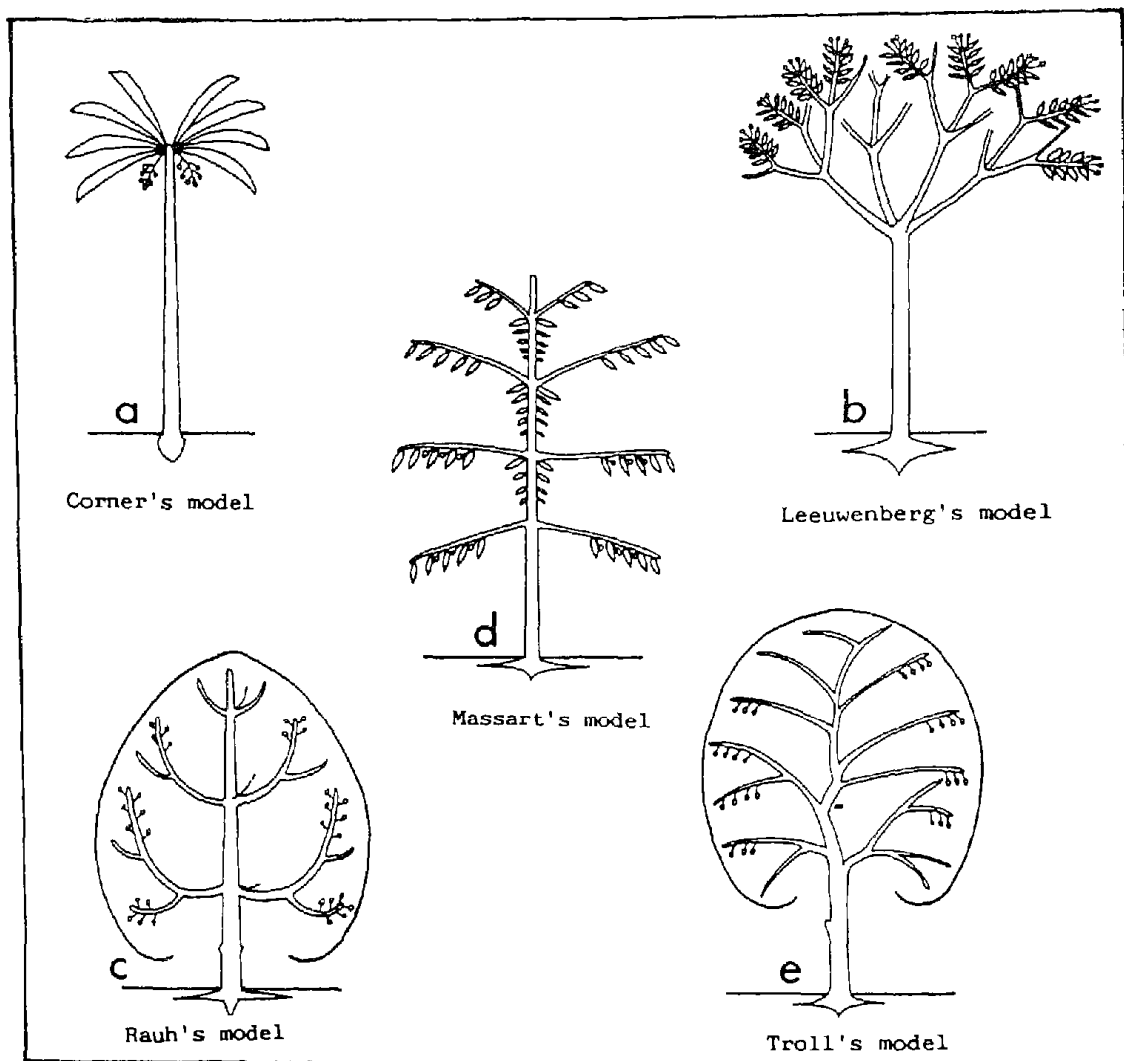


Figure 1. Architectural models.

conventional concept of habit since several models can exhibit a similar shape even though they function differently.

The hierarchy existing between axes clearly illustrates the concept of branching order (figure 2) in monopodial trees, studied in particular by Edelin (1984). The trunk is, by definition, an order 1 axis and the axes it bears are order 2 axes, etc. The different axes of a given tree have specific morphological features and can thus be grouped into categories. The description of all the categories of axes of a tree belonging to a given architectural model represent its specific architectural unit. These categories of axes are often linked to the concept of branching order.

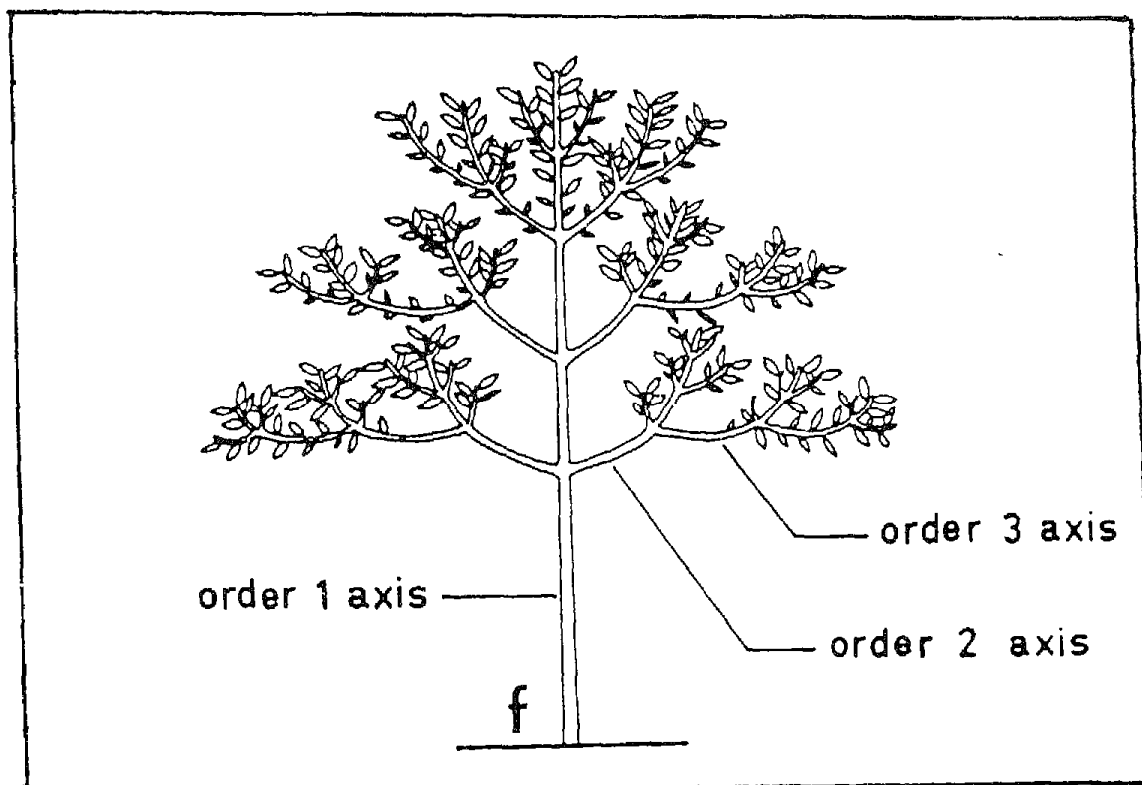


Figure 2. The concept of branching order.

The architectural unit of a tree does not usually allow it to grow past a certain stage. After the tree has reached a certain size it may duplicate its architectural unit according to a specific inherent strategy, referred to as “reiteration” by Oldeman (1974) (figure 3). A tall tree is thus a stack of reiterations, each of them representing a repetition of the architectural unit.

3. Stochastic processes for modeling the functional features of meristems

3.1 Qualitative description of meristem structures and functional features

Quantitative modeling can only be based on numbers characterizing measurable events. Qualitative analysis, as described earlier, enables grouping of tree axes into homogeneous populations, and provides an essential starting point.

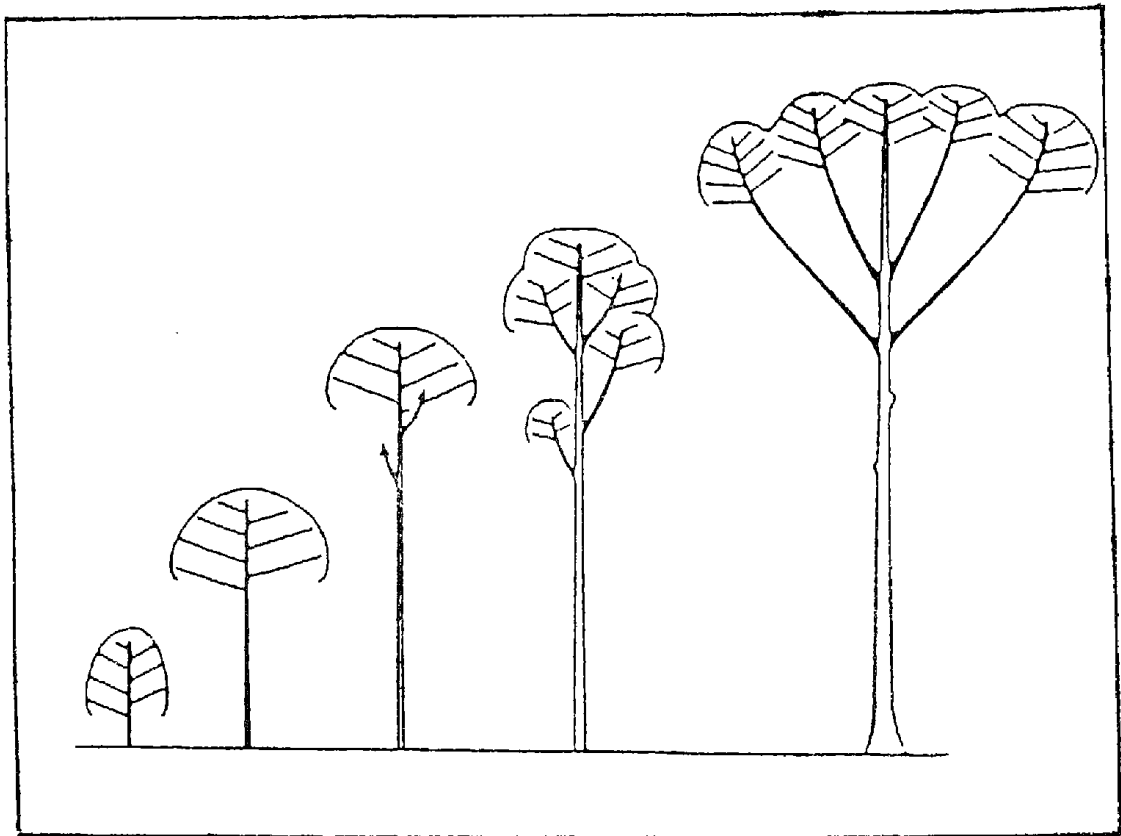


Figure 3. Growth and reiteration.

For us, the basic event is the appearance of a new internode at the tip of the leaf axis and its associated node which bears the axillary elements (buds, inflorescences). This event results from terminal meristem activity and it characterizes the growth process.

The axillary buds formed in the leaf axis can remain dormant during a certain period of time. Their development into an axis marks the branching process.

During the final phase, the meristems die, and the axes undergo self-pruning. This is the mortality process. Growth, branching, and mortality are the three processes that affect the life of meristems.

Their qualitative formation, described in detail by Rivals (1965), must be observed to model their functional patterns.

Minute embryonic leaves, separated by very short internodes, are produced in the buds by the terminal meristem according to a certain rhythmicity (apical growth). These productions, produced in the bud by organogenesis are referred to as the preformed part.

The production of new elements is invisible to the observer. Only the elongation and unfolding of the stock of preformed organs (internodes and leaves) can be seen (internodal growth). This elongation may affect some or all existing preformed organs. Elongation is limited by organogenesis. These two types of activity are not necessarily synchronized. The portion of axis that separates two growth cessations of the meristem is referred to as a growth unit (GU).

3.2 Random characteristics of meristem activity

This morphological description of terminal meristems corresponds to queueing theory. Apical growth is, by analogy, the law of customers' input; the stock of internodes produced, those who are waiting in line to be served; elongation, the system's law of customers' output.

In practice, only the law of output can be observed. Experimentally, this law appears to be random. Therefore, if the internodal growth of a population of leaf axes, measured in internodes (belonging to the same clone of the same age), is observed during a given period, we notice that the latter are distributed according to characteristic bell-shaped curves (figure 4a). Experimentally, it is impossible to predict the number of internodes the meristem will lengthen ahead of time.

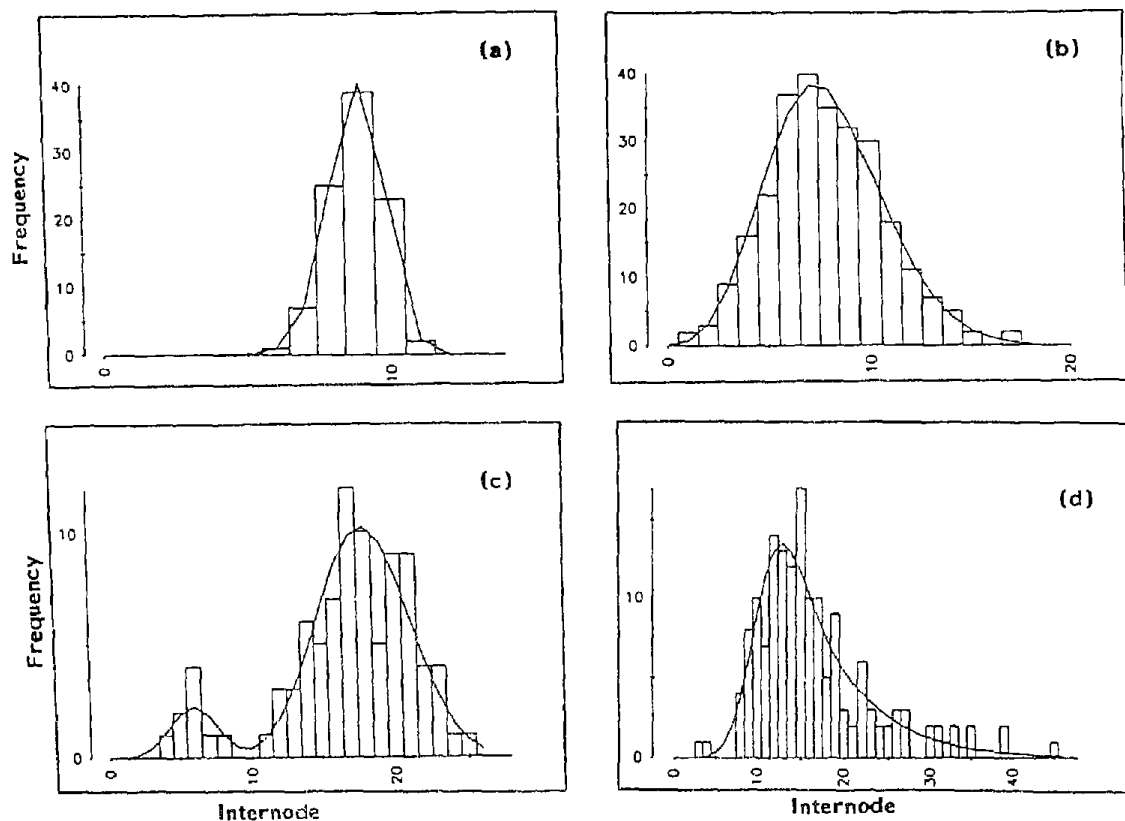


Figure 4. Statistic evolution of number of internodes/GU. (a) Young cocoa tree. (b) Litchi tree. (c) Wild cherry tree. (d) Apricot tree. (\square), Frequency; (—), internode.

It is also important to point out that plants are highly dependent on their environment. The development of the aerial structure depends on the soil, root system, and atmospheric interactions with the canopy and the environment.

Therefore, the parameters measured are essentially related to exogenous conditions. This phenomenon makes it possible to compare the functional pattern of a given plant under different cropping conditions so as to optimize these conditions.

4. Mathematical model and its validation

4.1 Poisson process and its breakdown

When there is a sufficient stock of preformed internodes, two successive internodes can appear almost simultaneously. When this is not the case, a minimum time interval is necessary for production and elongation of the internode.

The time interval between two successive leaves is generally said to be distributed according to a random function, the first two moments of which are μ and σ .

If the function is an exponential law, the law of occurrence at time t is a Poisson distribution. Feller (1950) showed that by using the central limit theorem, the law of occurrence converges to a normal distribution where:

$$\begin{aligned} \text{the average is } m &= t/\mu \\ \text{and the variance is } V &= t\sigma^2/\mu^3. \end{aligned}$$

For longer time intervals between successive leaves with $V < m$, convergence to a normal distribution and the integer value of the number of internodes ensure good convergence towards a binomial distribution (N, b). Therefore

$$Nb \simeq t/\mu \text{ and } Nb(1-b) = t\sigma^2/\mu^3$$

are good approximations. N corresponds to the number of artificial growth tests of the meristem called the “dimension”. The associated probability b is called “meristem activity” and $1-b$, “rest probability”.

It should be pointed out that in all cases observed in plants the discretization of time in regular intervals, characteristic of binomial growth, closely approximates the continuous growth process. Moreover, it simplifies the simulation methods (the Poisson distribution can be approached by the binomial distribution).

4.2 Interaction between the growth and cessation processes

When the growth process of a population of homogeneous leaf axes is interrupted at the same date, the internodal distribution can be observed directly (Poisson, binomial).

If the growth cessation periods are arranged according to a distribution, we will obtain a mixture arising from the combination of the growth distributions of the subpopulations interrupted at each cessation.

The final distribution has a probability

$$P(X=i) = \sum_{n=i}^{\infty} P(N=n) C_n^i b^i (1-b)^{n-i}.$$

4.3 Experimental observations of the number of internodes/GU

There are two main types of growth units from buds of the same physiological age in the trees examined:

GUs resulting from a single growth period of the meristems. The growth periods are of the same duration. The final distribution of the number of internodes

therefore results in a Poisson or binomial distribution (*e.g.*, litchi tree) (figure 4b). GUs resulting from several growth cessation periods during development. Several cases illustrate the meristem's aptitude to form a neoformed part after elongation of the preformed part during the springtime growth period (de Reffye *et al* 1991 b). When the elongation is accomplished, the elongation of the preformed part generally results in a binomial distribution.

Some of the meristems continue functioning as neoformations. Cessation of the neoformation in a tree can be almost simultaneous (*e.g.*, binomial neoformation law in wild cherry tree) (figure 4c), or staggered (*e.g.*, geometric neoformation law in apricot tree: in this case, the cessation probability is constant in neoformation) (figure 4d).

4.3a *GU metamorphosis in tree architecture:* Along a tree's natural botanical gradients (deflection, branching order, acrotony on GU), the neoformed part decreases as the preformed part increases. At the same time, the internodes get shorter. This phenomenon was measured in particular on the elm tree, wild cherry tree, and poplar tree, it characterizes the compulsory metamorphosis of a bud from the vegetative to the floral stage (figure 5).

In the trees that were studied, a progressive change of the parameters of the internodal distribution law is observed according to the physiological age of the GUs. Tree uniformity is due to the fact that this law applies to the entire tree, and characterizes its functional pattern.

4.3b *Appearance law for new GUs:* Appearance of new GUs is similar to internode elongation, but at a different scale. In tropical trees (litchi, rubber tree), the process from GU to GU is close to a binomial process. The coffee tree is a special case since its GU has only one internode. Binomial growth was studied for the first time with coffee trees.

The time interval between two successive GUs is variable: 15 days for the coffee tree, and up to several months for the litchi and rubber trees. For temperate plants, except for the juvenile polycyclisms, it is 1 year long.

The GUs are the visible marks of a tree's growth periods. In general, the separation between two GUs is marked by scars which remain visible for several years. For single trees, it is preferable to manage time according to the appearance of GUs.

4.4 *Branching process*

The new internodes produced by the terminal meristem bear nodes with axillary buds at the leafy axes. Branching can be instantaneous (syllaptic branching), or delayed (proleptic branching). So, there is a budding probability for dormant buds in terms of time. With syllaptic branching, the branching test is done only once. If it is not positive, it will never bud. With proleptic branching, the budding probability is tested at each dimension unit. In individualized GUs, the branching occurs in particular regions: terminal region, called "acrotony" (*e.g.*, poplar tree, apricot tree), and median region, called "mesotony" (*e.g.*, rubber tree). The branches can be grouped in packets along an axis and show significant correlations between

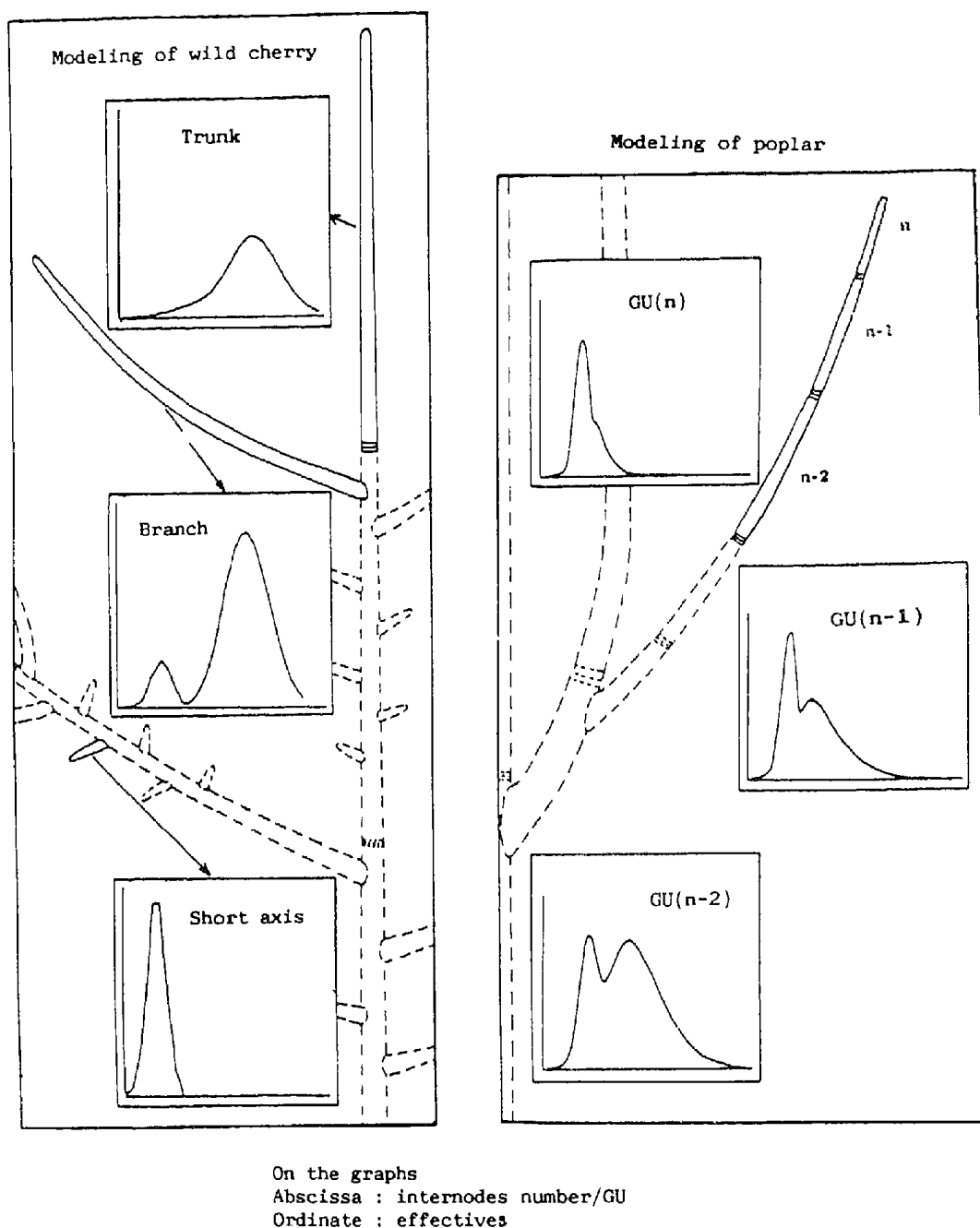


Figure 5. Distribution at different branching orders.

successive branches. Sequences of branches can be analysed by Markoff chains, thus the branching process can be measured like growth and mortality processes.

5. Reference axis

Studies of temperate crops and mature trees indicated the need to extend the concept of ramification order for describing a given plant. A mature order 1 shoot

may have the same behaviour as an order 3 or 4 shoot. Similarly, an axis with an order 3 or 4 behaviour can appear on an order 1 axis (figure 6).

The processes are thus described according to a unique, irreversible axis, termed “reference axis”. The reference axis translates the theoretical processes that affect a bud from the seed to the flower (de Reffye *et al* 1991a).

Laws governing bud activity are expressed as a jump along this axis. Growth is generally a short jump along the axis. Mortality—if it is not accidental—is a jump up to the terminal stage (flower).

The reference axis translates the notion of physiological age associated with a bud. It enables the description of growth strategies and overcomes the problem of reiteration. (A reiteration is branching with a small or no jump along the axis).

The concept of the reference axis, which is fundamental to the description of bud activity, can also be extended to the geometric properties of the elements resulting from growth (*e.g.*, angle of insertion of an axis, phyllotaxy, length, wood elasticity).

5.1 Simulation techniques

The simulation program is the same whatever the tree species is and is based on the reference axis and age data of the plant. It is divided into two parts: the topological part describes bud activity and the geometric part defines the spatial characteristics of bud activity.

Bud activity is simulated by the Monte-Carlo process in which random numbers are combined with stochastic laws of mortality, pause, and branching.

Buds are processed in prefix order according to this simplified pseudocode:

```

While there are buds to process, do
  Bud viability test
  If bud is still alive, then
    Bud growth test
    If bud is growing, then
      Calculate bud geometry
    End if
    Bud branching test
    If bud is branching, then
      Create a new bud
      Process new bud
    End if
  End if
End if

```

In each test a random number is checked against the law governing the process at the current stage of the reference axis. Each test could produce a jump along the axis.

The geometry of each element formed from a bud is computed in two steps:

```

If element is the start of an axis, then
  Calculate axis orientation in relation to main axis
  Calculate effect of weight on new axis
End if
Calculate local deviation of element

```

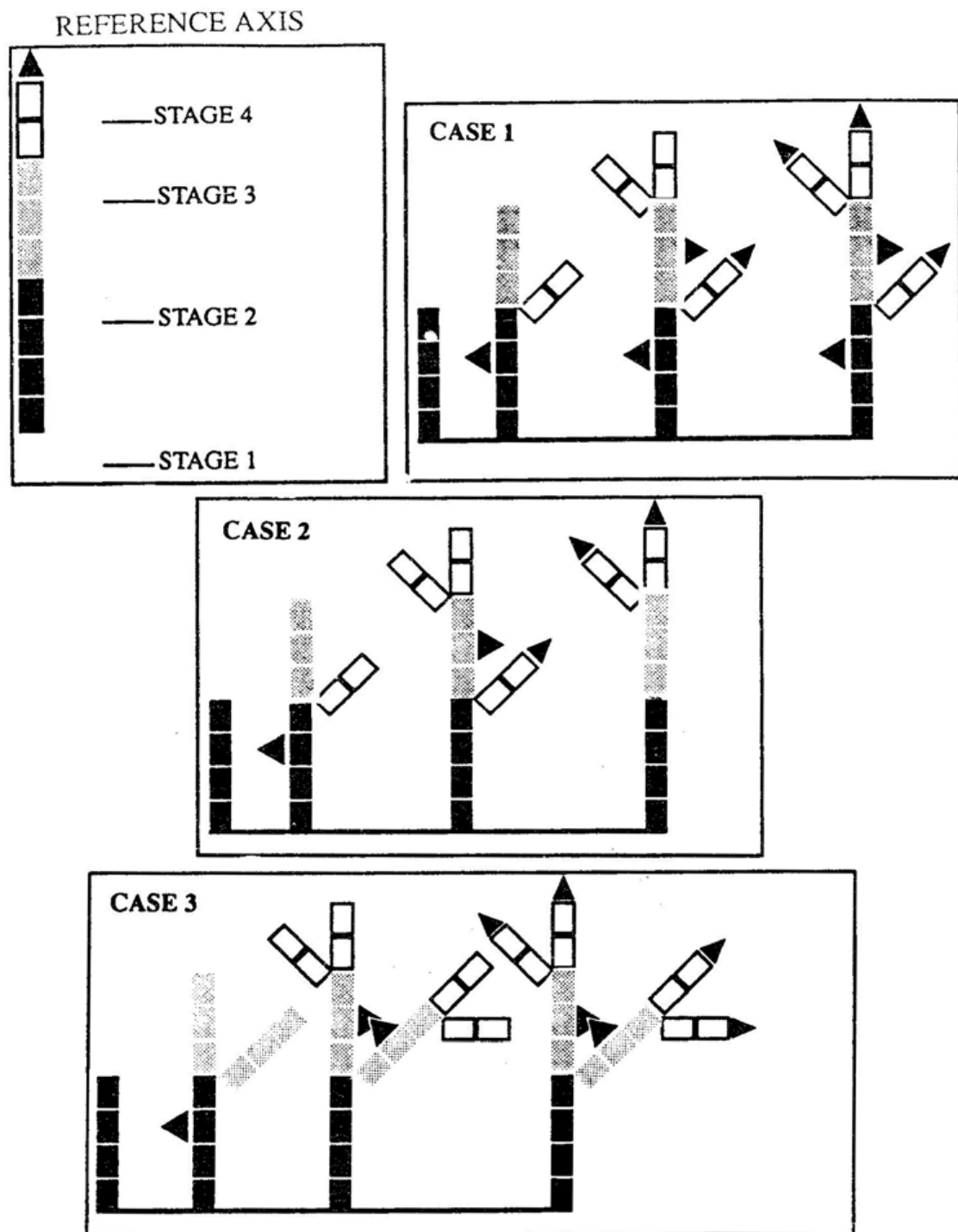



Figure 6. Growth principle expressed by means of a reference axis in the case of a theoretical plant showing 4 successive differentiation stages.

Case 1. Growth based on reference axis in the case of a plant presetting no self-pruning.

The trunk covers the whole reference axis step by step.

The jump made by shoots is dictated by an acrotonic gradient.

At every branching area the upper bud takes a jump up to stage 4.

When an axis reaches the last stage of the reference axis, plant stops growing.

Case 2. Same kind of process as in case 1, except that each axis undergoes self-pruning when reaching the end of the reference axis.

Case 3. Self-pruning occurs as in case 2, but the upper bud takes a local jump (reiteration) at every branching area.

Calculate scales (length, thickness)

Write result

The result of the simulation is a list of 4, 3 matrices describing the position and orientation of each simulated element. The simulation provides supplemental information such as the number of axes and flowers.

The simulation software is able to calculate the architecture of any plant defined by its reference axis parameters. Simulations based on different sequences of random numbers can also be compared. They reproduce the variability observed within the same species.

5.2 Structure of the simulation-based graphic database: the line tree

The simulated topological elements are stored in a line tree file designed by Jaeger (1987). The line tree contains all the information necessary for the representation of tree organs: transformation matrix, age of organs, symbol number, etc. It can be manipulated on a high level. It enables conventional sorting and geometrical operations. The tree can be visualized by linking the line tree and a 3-D base of organs (leaves, internodes) on the display (figure 7).

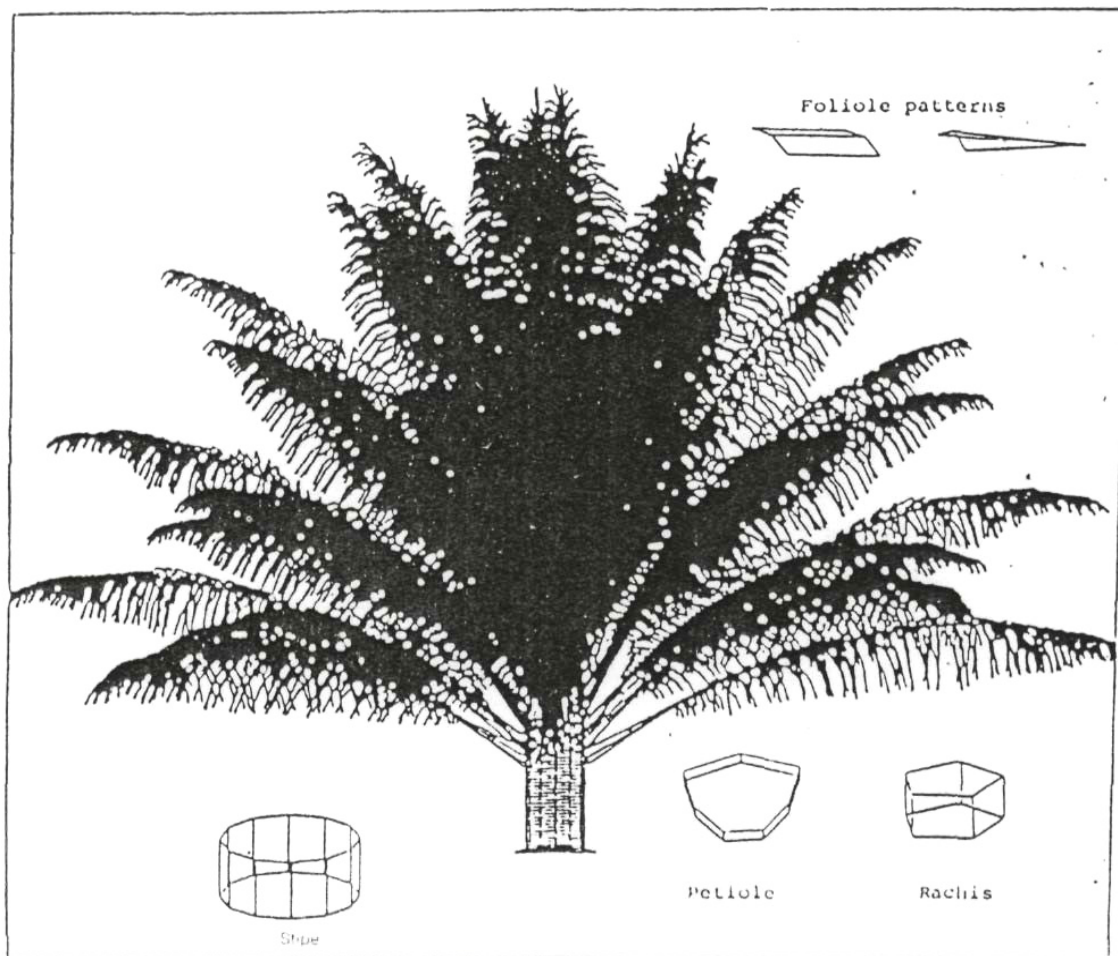


Figure 7. 3-D organs for the oil palm.

6. Trees: From the field to the computer

In practice, the simulation is carried out in several successive stages:

- (i) Botanical field observations provide accurate qualitative data concerning the architecture of a plant and its growth pattern.
- (ii) The plant is then numerically measured. The secondary axes are identified and described from the top downward according to their location along the main axes. Plant follow-ups are carried out. Distribution of the number of internodes/GU is recorded, as well as the location of the branches and meristem mortality. Geometrical measurements complete the data (length, angles of insertion, diameter, etc.). A database may then be created and interrogated to obtain statistics for events corresponding to a physiological age determined by the meristems.
- (iii) The third phase is the creation of a reference axis which is based on the different laws obtained from the trees and on their architectural evolution.
- (iv) The last phase is tree simulation on a graphic workstation. If the tree has been analysed correctly, the simulations should be very similar to the trees studied. A parameter file is then established which can accurately reproduce how this tree will develop in its environment.

7. Scientific field of application

Several sectors are concerned with applications of plant modeling: training, agronomy, landscape architecture and computer graphics. Development and architecture could be made attractive and educational if animated films were used



Figure 8. (a) Three random simulations of cotton tree. (b) Simulation of density variation on the cotton tree.

to illustrate how a tree occupies space. Films of this type are produced at the CIRAD Modeling Unit using stop-motion films of plant growth simulations of coffee, wild cherry, Japanese elm, litchi and other trees. The visualization of the concepts defined by Hallé (architectural models, reiteration) helps to understand them.

The quantitative aspect of this modeling makes it particularly easy for applications. The plant is expressed locally by statistical distributions which characterize the random functions of actual events (formation of internodes, branches, etc.). These random functions express the interaction between the plant and its environment (figure 8). If, for example, the environment is modified (by controlled irrigation density agronomical experiments), its impact on the plant development can be directly and accurately visualized. Different controlled situations can be interpolated to obtain optimum cropping conditions.

Other types of applications are also possible: identification of yield factors (number of flowers produced, leaf area, wood volume), study of the genetic variability for the architecture of a species, and light interception by leaves.

The CIRAD Modeling Unit has studied industrial tropical and temperate crops.



Figure 9a. Begonia flower.

Simulations by AMAP (Atelier de Modelisation de l'Architecture des Plantes) software produced by the Unit are shown in the following figures:

- oil palm (figure 7)
- begonia flower (figure 9a)
- growth of wild cherry (figure 9b)
- apricot (figure 9c)

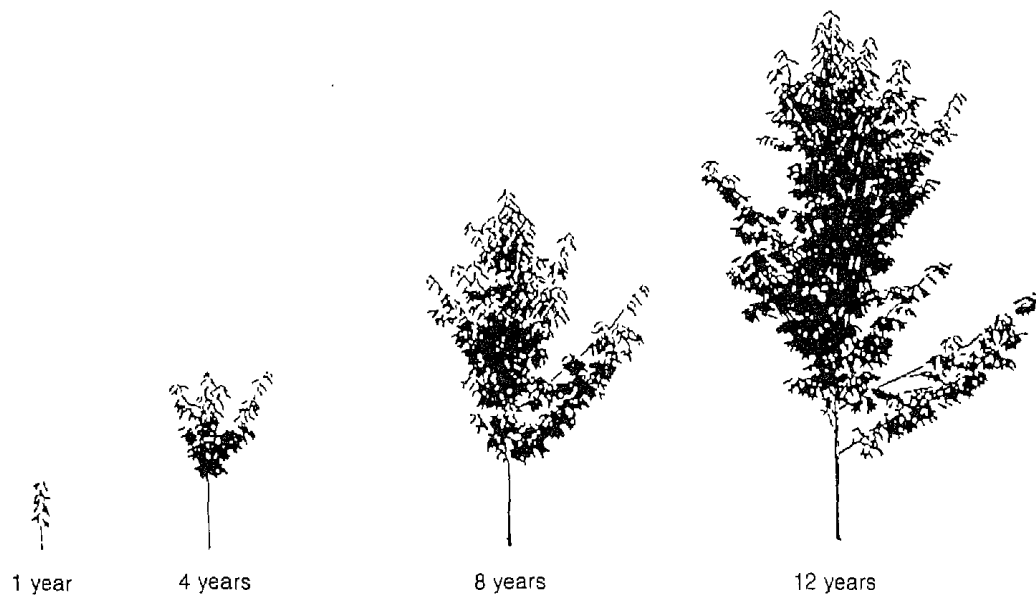


Figure 9b. Growth of wild cherry tree.

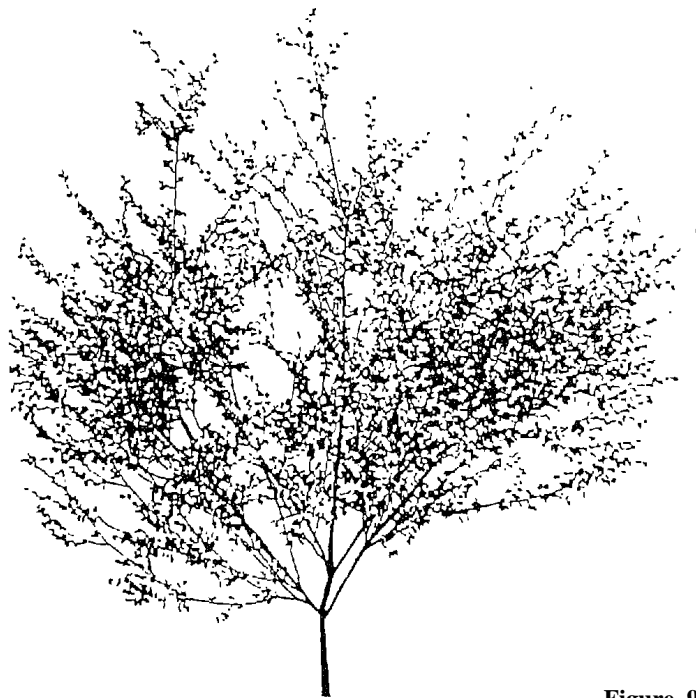


Figure 9c. Apricot tree.

- Japanese elm (at 2 years and 100 years) (figure 9d)
- Hevea (figure 9e)

For CAD applications AMAP is a useful tool for composing natural landscapes with seasonal changes. With the AMAP program, plants can also be selected for landscape architects. Parks, gardens, and other open spaces in urban centers can be

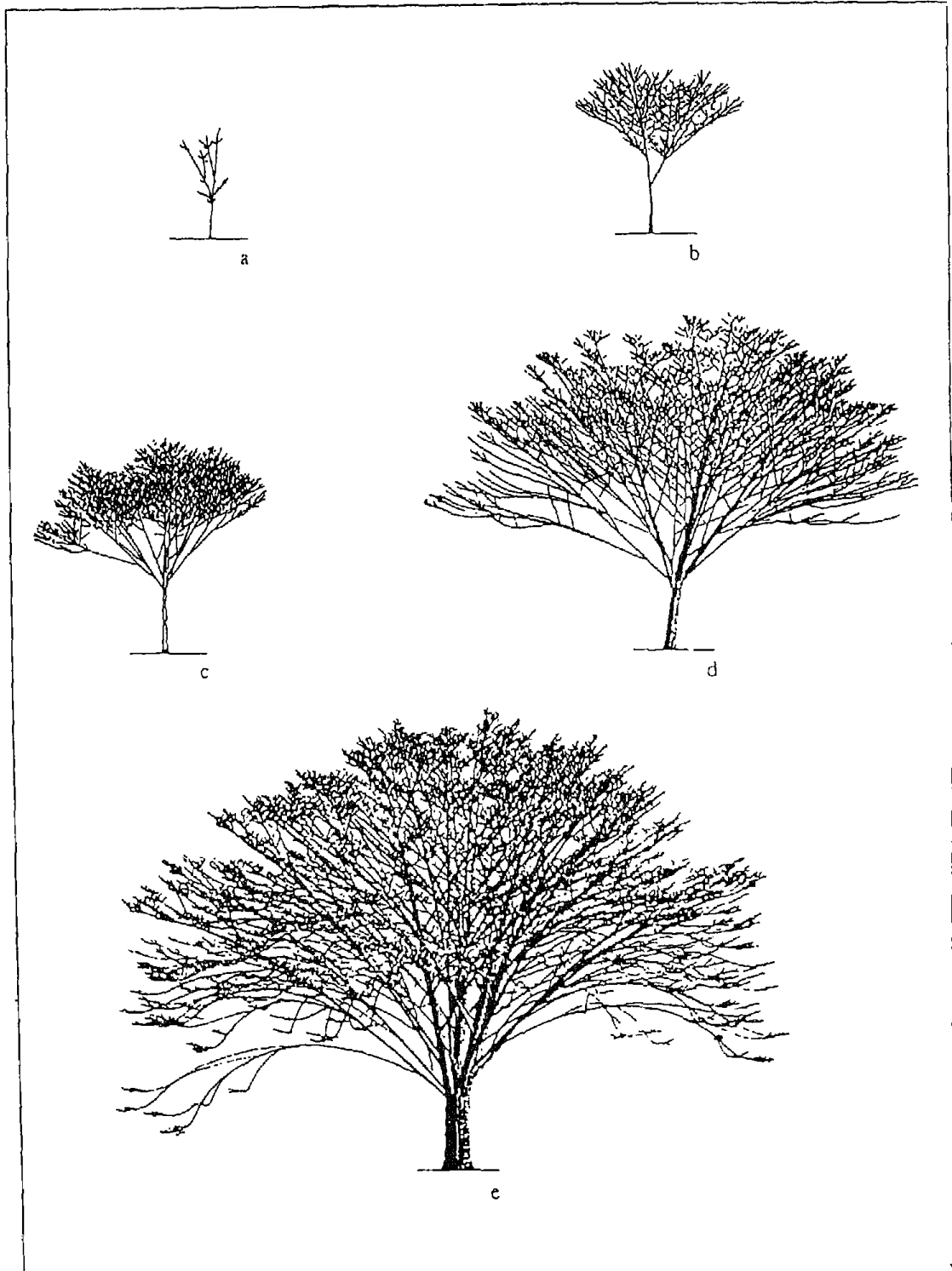


Figure 9d. Japanese elm.

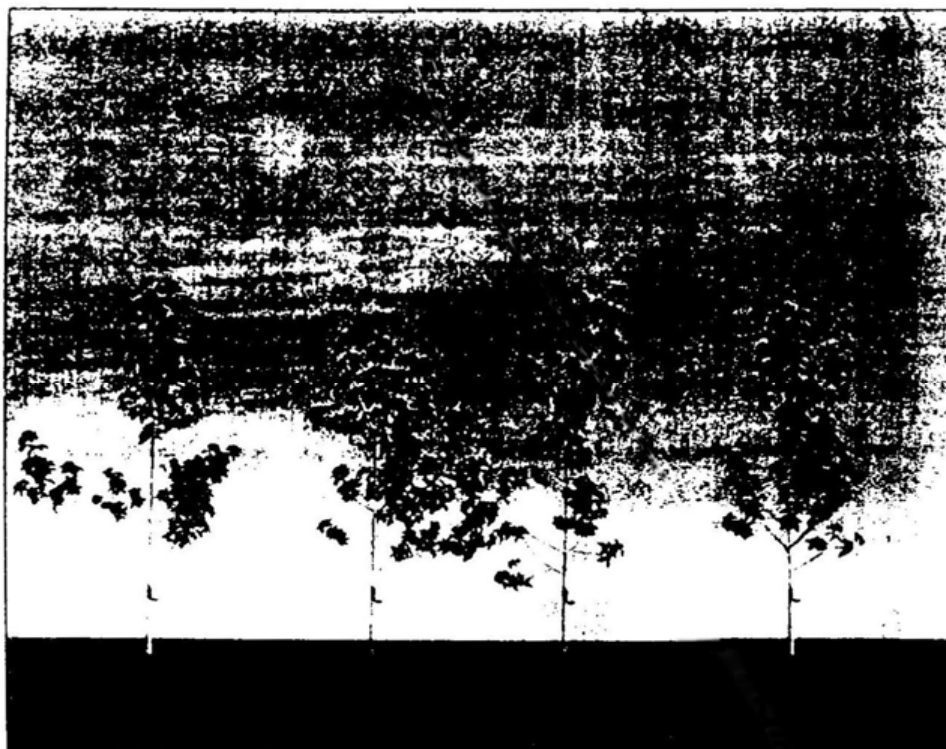


Figure 9e. Four simulations of hevea.

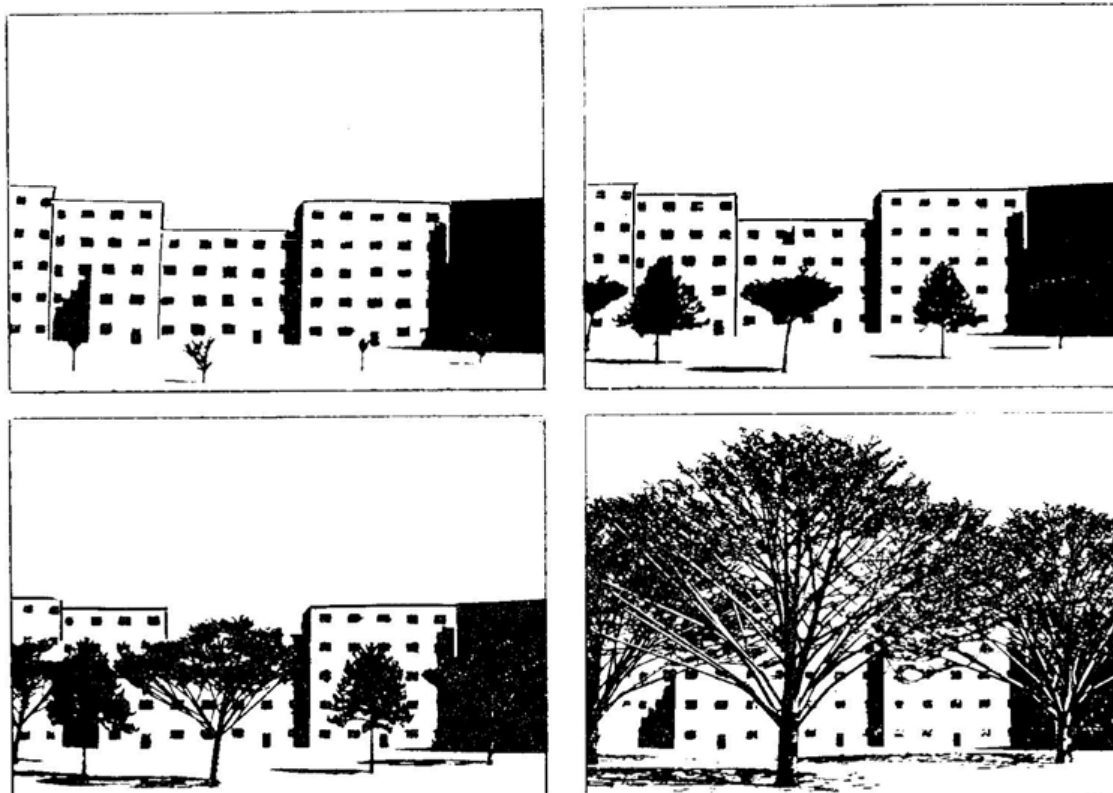


Figure 10. Application to landscaping.

(Figures 1- 10 were designed by D Barthélémy, E Elguero, M Jaeger and M Gérenton of the CIRAD Modeling Unit).

designed and their development predicted. This tool is quite helpful for landscape architects because it allows them to find the best solutions and avoid mistakes (figure 10). The AMAP software is also equipped with a regularly updated library of commonly used plants (trees, shrubs, flowers, grasses, etc.). It is easy to integrate in existing CAD programs.

AMAP has brought plants—real or imaginary—into the world of computer graphics. It offers live procedural objects for computer graphics, and contributes to the set of objects that modelers can use for scenario design.

References

- Aono 'M and Kunii T L 1984 Botanical tree image generation; *IEEE Comput. Graphics Appl.* **4** 10-33
- Edelin C 1984 *L'architecture monopodiale: l'exemple de quelques arbrres d'Asie tropicale*, Doctorat és Science thesis, Université des Sciences et des Techniques du Languedoc, Montpellier
- Eyrolles G, Francon J and Viennot G 1986 Combinatoire pour la synthèse d'images réalistes de plantes; in *Actes du deuxième colloque Image, CESTA* 648-652
- Feller W 1950 *An introduction to probability theory and its applications* vol. 1, 3rd edition (New York: Wiley International)
- Gardner G 1984 Simulation of natural scenes using textured quadric patches; *Comput. Graphics* **18** 11-20
- Hallé F, Oldeman R A A and Tomlison P B 1978 *Tropical trees and forests—An architectural analysis* (Berlin: Springer)
- Jaeger M 1987 *Representation et simulation de croissance des végétaux*, Doctorat thesis, Université Louis Pasteur de Strasbourg, Strasbourg
- Kawaguchi Y 1982 A morphological study of the form of nature (Proc. SIGGRAPH '82); *Comput. Graphics* **16** 223-232
- Lindenmayer A 1976 Paracladial systems; in *Automata, languages, development* (eds) A Lindenmayer and G Rozenberg (Amsterdam, New York, Oxford: North-Holland) pp 57 – 73
- Oldeman R A A 1974 *L'architecture de la forêt guyanaise* (Paris: Orstom)
- Oppenheimer P E 1986 Real time design and animation of fractal plants and trees (Proc. SIGGRAPH '86); *Comput. Graphics* **20** 55-64
- Prusinkiewicz P, Lindenmayer A and Hanan J 1988 Developmental models of herbaceous plants for computer imagery purposes; *Comput. Graphics* **22** 141-150
- Prusinkiewicz P and Hanan J 1989 *Lindenmayer systems, fractals, and plants; lecture notes in Biomathematics* 79 (New York; Springer)
- de Reffye Ph, Edelin C, Francon J, Jaeger M and Puech C 1988 Plant models faithful to botanical structure and development; *Comput. Graphics* **22** 151-158
- de Reffye Ph, Dinouard P and Barthélémy D 1991a *Modélisation et simulation de l'architecture de l'Orme du Japon *Zelkova serrata* (Thunb.) Makino (Ulmaceae): la notion d'axe référence*; (ed.) C Edelin (Naturalia Monspeliensia, n° h.s.) pp 251-266
- de Reffye Ph, Elguero E and Costes E 1991b Growth units construction in trees: a stochastic approach (Proc. 9th seminar of the theoretical biology school, Solignac (Fr) Sept. 1989); *Acta Biotheor.* **39** 325-342
- Rivals P 1965 Essai sur la croissance des arbres et sur leurs systèmes de floraison: *Proc. Journées d'agriculture tropicale et de botanique appliquée* **12**(12) Dec. 1965 pp 665-686, **12**(1-2-3) Jan.–Mar. 1966 91-122, **15** Jan.–Feb. 1967 67-102
- Smith A R 1984 Plants, fractals and formal languages; *Comput. Graphics* **18** 1-10

3D stem reconstruction from CT scan exams.

From log external shape to internal structures.

M. Jaeger¹, J.M. Leban², P.Borianne¹, S. Chemouny¹, L. Saint André².

Abstract

Non invasive medical imaging techniques gets more and more involved in wood quality. This work is a contribution to an EEC³ project and points out that post processing of CT scans on Norway Spruce logs provides a powerful tool for inner log structure investigation.

CT scanning on fresh Norway Spruce was done by Lulea University of Technology(Sweden). Exams were processed with the “C2000” software developed by CIRAD.

Axial log sections were treated to extract bark, sapwood and hardwood tissues. Specific tools were developed to generate parametric description of these areas. Specific studies were also carried out, mainly using new non linear filtering algorithms for ramification pattern recognition, pith location, ring counting.

3D reconstruction and visualization were performed on original and processed images. Realistic external shapes as well as internal structures shapes were obtained by merging several viewing techniques.

Results gained through this process lead to an accurate geometrical description of bark, sapwood, hardwood, branches boundaries and pedagogic 3D visualizations. Moreover, when including pith location, a wide range of uses is possible from resource evaluation to sawing simulation. It also contributes to define validation tools for external log shape description approaches.

CR Categories: Computer graphics, image processing, volume visualization, geometric modeling, medical imaging.

Keywords: CT scan, log, Norway Spruce, 3D reconstruction, 3D visualization, pattern recognition, wood structure, bark, ramification, hardwood, sapwood.

1. Introduction.

Forest resource evaluation as well as log to end user products conversion simulations ask for faithful external and internal properties evaluations [Leban99]. If image vision is today a classical way for log grading [Grace94], [Mongeau93], its use for internal diagnosis has not reach yet the industrial scale although significant research results were gained in the last past years [Wang97][Grundberg97][Oja97][Hagman99]. Non destructive image acquisition gets more and more involved, log CT scanning brings really an accurate new way to explore them. Variation of X ray response in transverse scan images reflects tissues changes or/and moisture contents changes at an interesting level of spatial resolution.

The aim of this work, contribution to CT96-1915 EEC “STUD-PPP” FAIR project, is to process Norway Spruce CT Scans exams in order to extract the geometry of the main wood structures in the logs. This work is done on the basis of the C2000 software [Jaeger93][Canovas95][Treil95][Joyeux96][Prudhomme98], developed at Cirad for medical purposes. Specific modules were of course developed for this application in order to perform automatic image processing dedicated to various wood tissues recognition. Adapted 3D reconstruction methodologies were also defined.

Two main kinds of data set were used: gray level images coming either from the original CT scanned images either from processed CT images; labeled images built from processed images. These last images are used to generate parametric geometrical outputs.

¹ CIRAD MEB Unité de Modélisation des Plantes BP 5035 F-34032 MONTPELLIER Cedex 01

² INRA-ERQB-F-54280 CHAMPENOUX

³ STUD Fair CT96-1915 EEC project “improved Spruce timber utilization” .

1.1. Process data flow.

Process data flow can be summarized as following steps:

- CT images are received, converted and compressed
- Gray level images are processed,
- Labeled images are computed,
- 3D reconstruction and visualization is performed with previous images,
- Additional external inputs are given to generate the parametric geometrical outputs.
- The geometric data is used by the resource evaluator or by the sawing simulator.

Following data flow chart shows used data and processes (Fig 1.1)

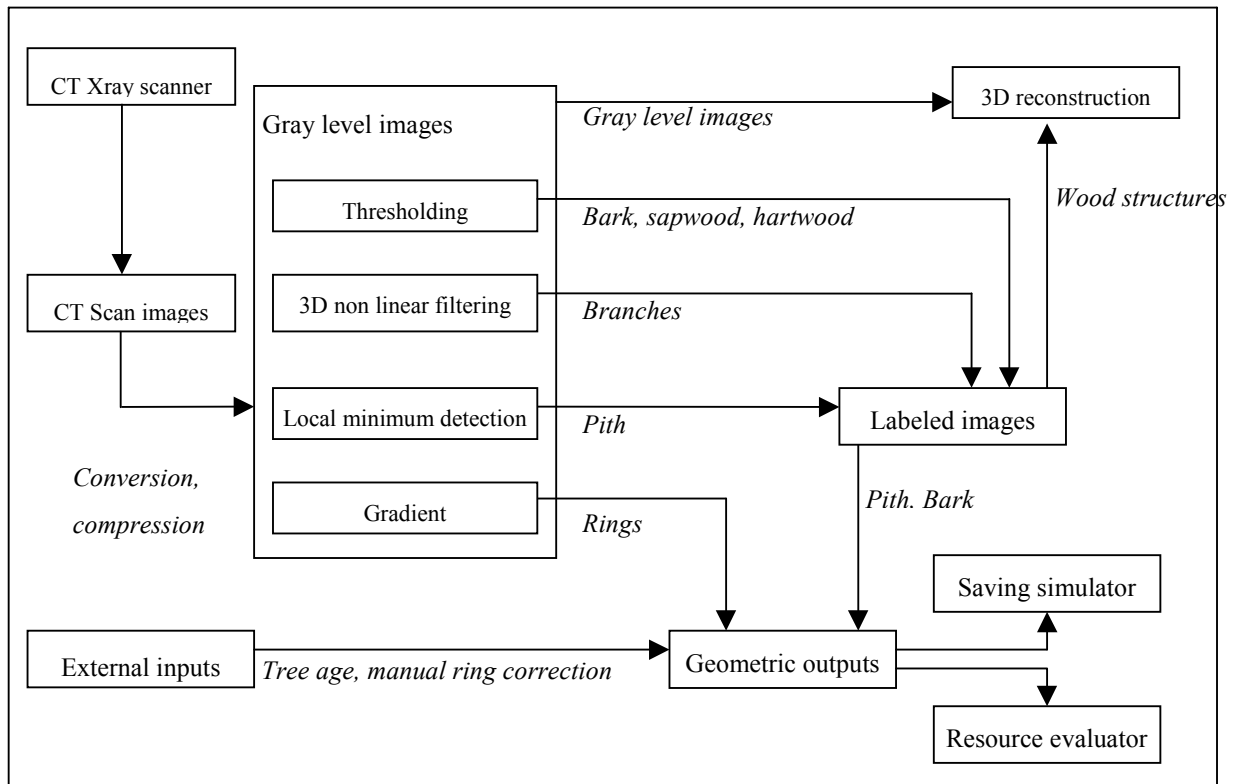


Figure 1.1. Log CT scan process data flow chart.

1.2. Overview.

In this paper we will first briefly expose the acquisition protocol and the exam conversion, then describe the process used to extract bark and sapwood from the images. We will then focus on the branch detection and the pith localization. In a fourth step, 3D reconstruction and visualization are described. Then processes used to obtain geometrical outputs will be presented. Finally, applications and process performances will be reported.

2. Image acquisition and conversion.

2.1. CT scan protocol

Norway Spruce log CT scanning was performed by Lulea University of Technology (LTU) in Skelleftea (Sweden)[Grönlund96]. Original reconstructed images from 5 logs were sent to Cirad (in Montpellier, France) on CD-Rom support or using remote FTP (File Transfert Protocol) access. These images are stored in their original format (Siemens in house file description).

Image protocol acquisition defined by LTU was fixed to 512 by 512 pixel per image for transverse cross section with a field of view (FOV) of cm and a slice thickness of 1 cm. Gray levels are encoded on 2 bytes for a typical range of 4096 CT numbers. Typical exam consist in 450 slices giving a storage volume of 240 Mb.

2.2. Image conversion and compression.

Data volume is a crucial point while processing. The classical way to decrease it is to limit pixel encoding on 1 byte (giving a range of 256 CT number values) and divide the image resolution by 2 on both axis. That is done in Lulea [Grundberg92] dropping down the storage volume to nearly 30 Mb per exam.

We chose to preserve full dynamic and resolution in the images. Dynamic range in the transverse CT scan images can be divided in several classes as follow:

- 0 to 200 air and noise
- 200 to 1200 wood tissues that can be subdivided in two sub classes:
 - 200 to 500 bark and hard wood
 - 500 to 1200 branches and sapwood

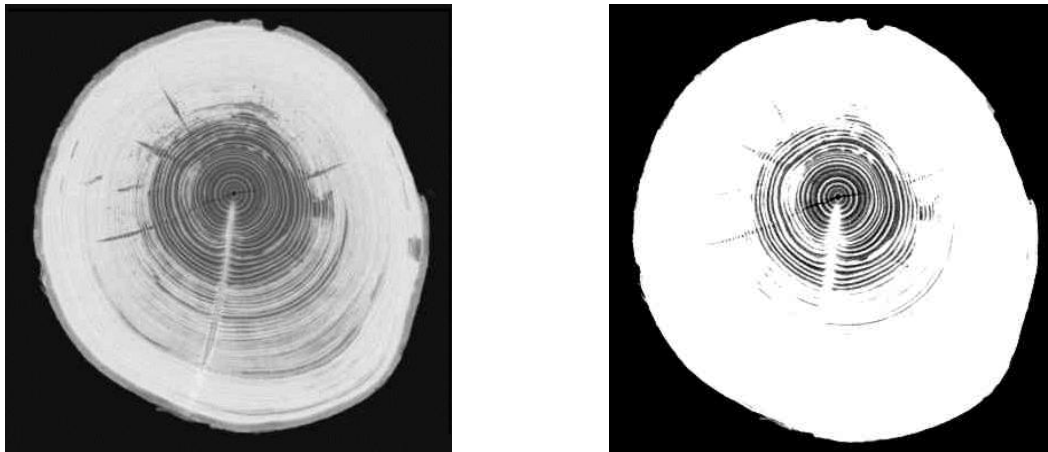


Figure 2.2. Cross section visualization. Right: all tissues 200-1200 CT value. Left: 500-1200 CT value.

In order to reduce the exam storage volume and to be processed in the “C2000” software images were converted in an in house image format as follow:

.threshold non interesting low CT scan values (to reduce noise in air areas, all CT number below 80 were set to value 80).

.encode CT values on 12 bits (3 bytes for 2 pixels, that doesn't affect the dynamic range).

.compress (with no loses) data using a single run length encoding (RLE) technique set of consecutive pixels that have same CT number value. This is really efficient within the part of the image that is outside the log (air) and that represents nearly 30 to 50 % of the whole surface of the image.

This conversion dropped down the storage volume from 240 Mb to nearly 60-70 Mb per Ct exam without losing any significant data.

3. Bark, Hardwood and Sapwood.

Single threshold according to CT number value is performed to detect these wood tissues. This work is done following these steps:

. Threshold the current image at value 200. This leads to select all pixels bellowing to the log. Keep the result in a labeled image those unique region is called “Log”.

. Threshold the same image at value 500. This leads to select all high CT number values : branches, sapwood and parts of hardwood (high CT number values due to rings). Fill this area, smooth it and name it “Solid Wood”. Overwrite then these labeled image to the previous one. Previous “Log” area is now limited to the external low CT number values, that say the bark. (Figure 3).

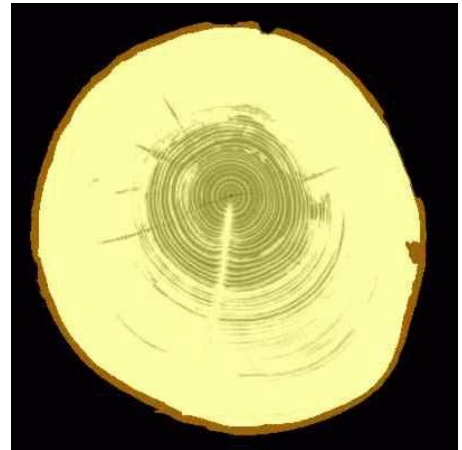
. Do the same process to separate hardwood and sapwood tissues with a new area obtained from low CT number values inside the “Solid Wood” region.

. Perform these steps on all slices using a automatic script.

Thus this process can be automatically applied on the full CT scan exam.

However, some trouble may occur according to the “quality” of the log. When parts of bark are not present or are permeable to the air on the section, underlying wood tissues can dry and therefore have a lower CT number value. (See the right side of figure 3). This can be corrected by locally smoothing the surface or with a manual editing tool.

Figure 3. Bark and solid wood structures.



4. Branch detection and pith location.

4.1. Detecting branches within spruce CT-Scan exams.

Because of the acquisition protocol (1 slice per cm) and the low thickness of the branches, a branch insertion is usually fully visible in only one cut (figure 4.1.a) . Its detection will be performed by differentiation on successive slices, i.e. we look for features that are visible within one specified cut and absent (at the same planar location) in the previous cut and in the next cut . To be efficient this differentiation has to ignore small changes between two cuts due to noise and rings variation, therefore all the slices are processed with a special in-house filtering (details are to be found in [Chemouny 99]) that enhances features such as straight lines while smoothing out noise and rings (figure 4.1.b) . Note that even if our filtering tools are intended to process 3D data set of voxel, the anisotropy of log scans (1cm between two slices vs. less than 1mm² pixel area within CT-plan) is too strong to allow an actual 3D filtering, furthermore we want to enhance the differences between two cuts and a 3D filtering would tend to weaken these differences . For these reasons the slices will be independently filtered with a set of 2D neighborhoods .

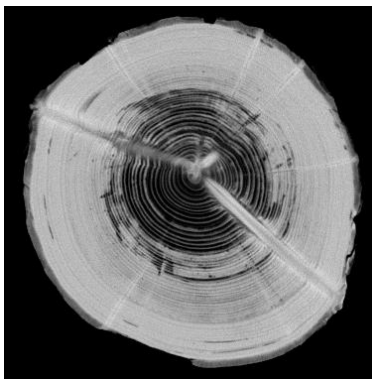


Figure 4.1.a

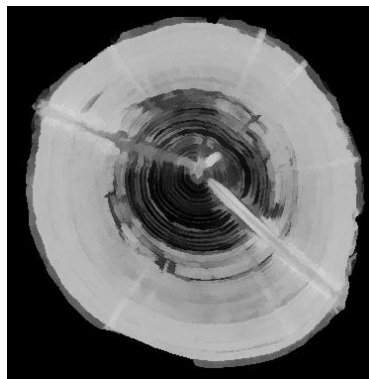


Figure 4.1.b

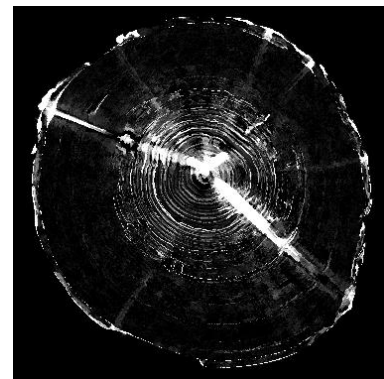


Figure 4.1.c

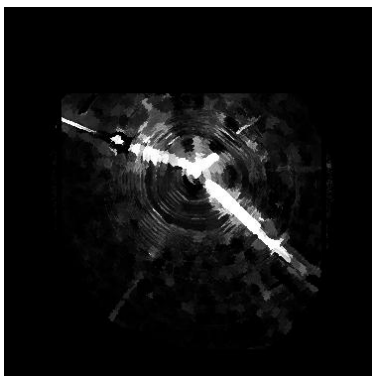


Figure 4.1.d



Figure 4.1.e



Figure 4.1.f

After this stage the slices are ready for the differentiation process . We assume that branches are significantly bright and straight features . For each slice an arithmetic difference is performed with its previous cut and an

other one with its following cut, negative values being set to zero (we're looking indeed for what appearing brighter in the current slice) . The "differentiation" image is then generated (figure 4.1.c) in combining both images coming out from the arithmetic differences, the (gray level) value of each pixel being the maximum value between the two corresponding pixels of those images . One can note in fig. 4.1.c that although branches are well detected, other details like bark or some inner rings are still present . Bark problem is due to strong variations of the external shape of the log from one slice to another. This issue is easily resolved when defining a region of interest (ROI) by "peeling" the log (the most external pixels of the log are removed), branches detection will be performed only within this ROI . Some inner rings are still present because they are the widest and the best defined in the original images, therefore the filtering has preserved them instead of smoothing them, thus the variation of these rings are significant enough to be observed . Since they are not actual rings but variations between rings (i.e. they are less clearly defined and thinner than original rings) they won't subsist after another filtering process . Results after the second (identical to the first one) filtering and the limitation of the image at the previously defined ROI are shown in figure 4.1.d . At this stage a 3D region growing (based on "scan line" algorithm and regions fusion according to their averaged gray values [Fiorio 95,96]) is applied (figure 4.1.e), it is efficient enough to allow a good visualization of the inner skeleton of the log, to count the number of branches, characterize the orientation of each branch, and detect all the whorls (figure 4.1.f) .

This method has been tested on numerous spruce CT-Scans (usually several hundreds of slices) and has found to be very robust . The results are very good in terms of detection (all the branches, even the very thin ones are detected) and computational time (less than 1 sec per cut) .

4.2. Detecting the pith location.

We have provided a fast and efficient near-automatic method for pith localization within spruce logs CT-Scans .

On such images we can observe that :

- the pith is locally the darkest pixel
- the pith location does not vary a lot (according to its planar coordinates) from one slice to another.

The idea underlying the pith localization is very simple, on a first cut the pith is manually placed, then automatically propagated to the following slice, its position is corrected in looking for the pixel of minimum value around the projected pixel (usually within a 5 by 5 area), the new pith is then projected to the next slice, and so on until the end of the log .

Problem may occur with very big branches that displace the pith outside the neighborhood of the projected previous pith, in that case the pith location has to be manually corrected and the process is restarted from this slice . In practice this problem occur less than 1 time over a hundred slices, so manual corrections are very rare, and the operator role is essentially to perform a control task . See Figure 5.1.

5. 3D reconstruction and visualization.

3d reconstruction and visualization is processed using several techniques. These techniques mainly differs by their respective inputs that can be gray level images (T1, T4) or labeled images (T2, T3) and by the way the rendering is processed: pixel per pixel or from edges (T1bis, T2bis). In these last approaches, edges are extracted from binary or labeled images and stored as discrete segment lines [Borianne96].

Technique T1 consists in a single 3D reconstruction of gray level images. Each pixel of the CT exam, in each serial image is converted in 3D coordinates according to its log position and just rendered with a depth map buffer. Only pixels those gray levels are visible an appropriate lookup table are rendered (black pixels are skipped). The output is a gray level image.

Technique T1bis is a wire frame 3D visualization. Input are also the gray level images. They are threshold according to a lookup table and converted to a binary image. Edges of these image are then transformed in 3D segments and rendered in white with a depth map buffer. The output is monochromatic.

Technique T2 is the same technique than T1 but with labeled images as input.: Each labeled pixel in each serial image is converted in 3D coordinates according to its log position and rendered with its area color and the depth map buffer. The output is a color image.

Technique T2bis is a wire frame 3D visualization of the labeled images. Labeled areas are rendered from their edges with their respective area color and a depth map buffer. The output is a color image.

Technique T3 is nearly the same than technique T1 and T2. For each pixel bellowing to a set of labeled areas, rendering is done with the corresponding gray level color instead of the area color. Labeled areas to be visualized are chosen by the user on a single list.

Technique T4 is a high resolution iso-surface reconstruction. Gray level images are threshold on a sub pixel level

using the dividing cube approach [Cline88]. Various threshold values are possible, a specific color, transparency and diffuse reflection coefficient is affected to each of them. High quality rendering is generated from the adaptive subpixel sampling and illumination with diffuse reflection computed from 3d gradient fields.

As illustrated bellow, these techniques can be merged on multiple pass image reconstruction and depth realism is gained adding a depth of field color correction (decreasing color intensity according to depth). Cut planes can be defined from the data set or from the viewer point of view.

Images reconstruction and visualization examples.

Figure	Techniques used	Specific process
5.1	T2 for the pith and branches, T2bis for the bark on labeled images	Depth of field correction added.
5.2	T1 with a 800 CT threshold value on processed gray images.	3D Contrast enhancement.
5.3	T4 on high CT numbers original gray images T1 on low numbers.	Transparency on low gray levels.
5.4	T4 with a 400 CT threshold value on original gray images.	Negative light illumination.

Figure 5.1. visualization of the bark, the pith and the branch emissions of a spruce log scan

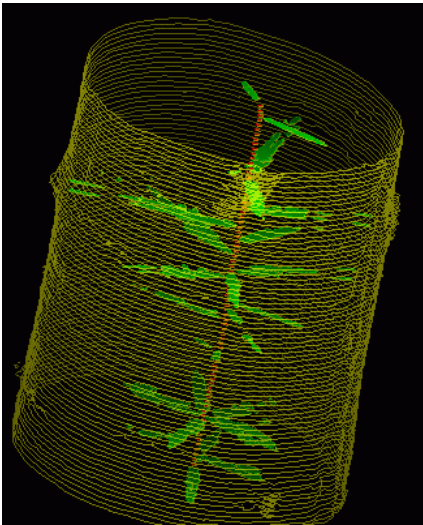


Figure 5.2. Part of half sawed log

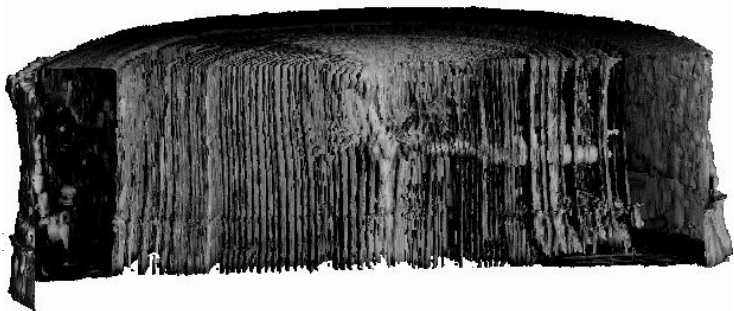
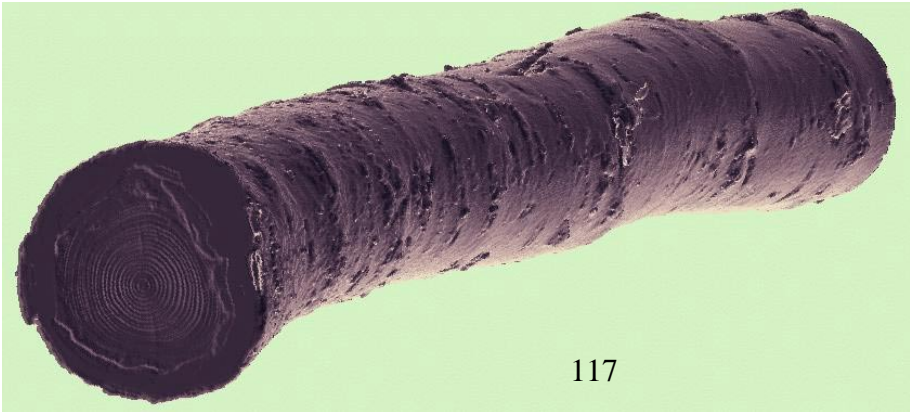


Figure 5.3. Half sawed log with sapwood, knots and transparent rings.



Figure 5.4. External realistic reconstruction



6. Geometrical output generation.

Geometrical output generation is performed from the labeled images. It mainly consist in a sector generator and various control fields.

On any section, pith position can be located by the user. In order to minimize human interaction, automatic interpolation is performed from 2 given user guided localization. As an option, automatic pith detection described in section 4.2 can be run to generate automatically these positions on each slice.

Sector are defined by a parametric number of rays starting from the pith position. Intersections between these rays and the various areas in the labeled images are stored as the geometrical description of the section (See the blue square landmarks in figure 6.1 for the external shape under bark description). This information is then completed with tree and log information through the “Wood-box” dialog panel. (See Figure 6.2).

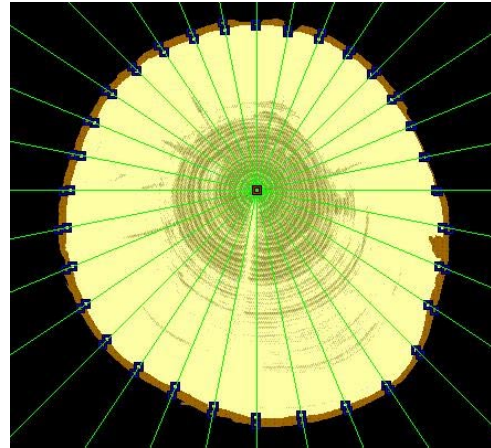


Figure 6.1 Generating the geometrical output.

Tree

number 31 age 60 h (cm) 285

Log

number 5 pith X 232.90 sectors 32
 - Y 285.60
 - Z 565.00

Growth units

number 43	from slice 114	to slice 117
number 44	from slice 118	to slice 120
number 45	from slice 121	to slice 124

☐ Selection

1/1 current [left arrow] [right arrow] bark

☒ Rings

Geometry
 Monotonous
☐ edit

☐ filter

Treatment

by step of 10 from slice 114 to slice 430

Apply Save Options Close

Graph: A plot of L (cm) vs L. The y-axis ranges from 0 to 3, and the x-axis ranges from 0 to 1. A red vertical line is at L=0.5, and a red horizontal line is at L=1. A black curve starts at (0,0) and rises to (1,1).

sectors	0	1	2	3	4
found r.	100	101	102	103	104
added r.	200	201	202	203	204

Figure 6.2 The “Wood-box” dialog panel in the C2000-software.

This dialog panel allows also the user to generate ring description. Gradient gray level images are generated as a support for ring detection. Then, starting from the pith, intersections from sector rays and high gradient pixels are stored. Specific algorithms try then to connect intersections coming from the same ring.

The methodology works quite well near the pith on the 15 to 20 first rings as shown in figure 6.3. This picture outlines high gradient values after local non maxima suppression [Deriche87].

Nevertheless, this approach is not really efficient on the other rings because of the poor spatial resolution and the small ring width, and need expert correction that is really time consuming. Once more, in order to speed up the process, the user can enter for a given slice the number of rings and the system will generate the missing rings according to a constant ring width or a curve designed ring width distribution. (See “Ring” part of the “Wood-Box” panel in figure 6.2).

External output can be generated for each slice or by steps as shown in the lower part of the panel.

Output files are designed as ascii files that can easily be read in statistical tools, resource evaluator and sawing simulators.

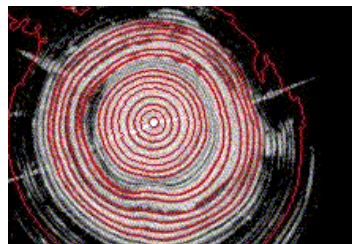


Figure 6.3. Ring extraction from gradient.

7. Performance and applications.

7.1 Hardware requirements.

The C2000 software range product is developed in C and C++ language under UNIX/OSF Motif environment for SGI, HP, SUN and DEC workstations and under Windows 95-98-NT environment for PC stations. Typical configuration is 64 Mb of RAM, 2 Go of disk, a display area of 1024 by 768 pixel under 256 colors. No specific card is needed.

Following figures are given for a low cost hardware platform like a Pentium II 233 Mhz machine without any optional graphical card (2 Mb video ram) a 2 Go disk and 64 Mb of Ram. All reconstruction images are build on 24 bits, non depending on the display specifications.

7.2. Space and time figures.

Exam	Nb of slices	Nb of images	Space on disk	Temporary disk space	Temporary Ram space
31	450	3600 (*)	60 Mb	450 Mb (**)	42 Mb (***)

(*) For each original gray scale image OI, 2 processed gray scales images PI1, PI2 are computed as well as a labeled image LI. Moreover, an icon image is built from each full resolution gray scale or labeled image.

(**) Gray scale images are encoded on 2 bytes and labeled images on 1 byte with a rough RLE technique. 3D reconstruction needs at least 16 Mb disk space for image and depth map temporary storage.

(***) Ram space is poorly depending on the number of slices. 3D reconstruction need at least 16 Mb ram space for a 700 by 500 image reconstruction for depth maps, transparency maps, colors and gradient maps.

Process	Figure	Volume	Time (user)
Image Conversion/compression	2.2	450 slices	105 s.
Bark, hardwood, sapwood extraction	3.	445 slices	200 s.
Branch detection	4.1- 4.6	440 slices	400 s.
Pith detection	5.1	445 slices	25 s.
3D reconstruction	5.2- 5.4	450 slices	20-200 s.

7.3 Applications.

CT image postprocessing allows three kind of applications:

- qualitative better understanding of the wood internal structures from accurate 3D reconstruction and visualizations, mainly thanks to transparencies and clipping planes.
- a realistic, non destructive, high quality characterization of various geometrical features of logs that allows studies on section anisotropy, section eccentricity, pith location variation, long bending, branch counting. That illustrates the interest of this approach to define quantitative figures on geometrical shapes and, in the future, on other wood property aspects like wood density, wood compression area, moisture content.
- The third kind of application of such tools is the use of its outcomes as input in resource evaluation and sawing simulations. This outcomes are really close to real shapes and can therefor lead to faithful volume estimations. It also brings a set of data strongly linking external and internal shapes. That is a very useful tool for model and software validations that try to define internal properties from the external shape [Saint-André98] and also for all the classical models that work on the basis of simplified section description (circular, elliptical in the best case).

Conclusion

We shown that original CT scan sets of fresh Norway Spruce logs from LUT can be efficiently processed with low cost workstations. Storage mass was dropped down without lose of any useful data. Adapted image processing techniques lead to automatic extraction of various tissues and anatomic features. Several wood quality parameters (i.e. knots, wood density variations, ring width) could be represented by realistic 3D reconstruction and quantified in terms of location and shape.

The quantitative geometrical outputs were tested for resource evaluation, sawing simulation. On a larger scale, the systematic image processing of the Spruce European data base collected through the STUD research project (data base built by Prof. Anders Gronlund's team) would produce very detailed data sets in order to analyze the relationships (*i.e.*, models) between the external log shape and the internal wood properties with a very low sampling cost (images are available and it is not necessary to perform a destructive sampling including sawing logs, measuring the knots attributes, grain angle and ring width on the boards etc.).

On one hand, such relationships or models could be hopefully implemented for the optical scanners, which are only able to record the external log shape.

On another hand, the full 3D representation of the scanned logs including their internal features, would contribute to a better understanding of the timber properties variations within logs for Professors, researchers and also for the sawmill managers.

So, post processing of CT scans provide a powerful tool for any non destructive inner log structure investigation. Future work will now be focused on specific tissues detection and characterization like wood compression area and internal defects (resin pockets, cracks, ...). This work could also be extended on other species like oak.

References:

- [Borianne96] Borianne P., Jaeger M. **1996**. Polygonisation reversible en imagerie médicale. Application à la visualisation de divers tissus anatomiques. *In: Actes du 3^{ème} Colloque Africain sur la Recherche en Informatique CARI'96*. Libreville (Gabon), 9-15 octobre 1996, pp. 114-123.
- [Canovas95] Canovas F., Prudhomme M., Jaeger M., Bonnel F. **1995**. Three Dimensional reconstruction of the wrist biometry of the carpal bones. *In: 4th European Congress of the European Association of Clinical Anatomy*, Vienne (Autriche), sept. 1995, p. 192.
- [Chemouny99] Chemouny, S., Joyeux, H., Masson, B., Borne, F., Jaeger, M. and Monga, O., **1999**. Advanced 3D image processing techniques for liver and hepatic tumors location and volumetry. *In: Medical Imaging 1999 - Image Processing*, K.M. Hanson (Ed.), San Diego, USA 20-26/02/1999. SPIE The International Society for Optical Engineering. *SPIE Proceedings Series*(3661-1); pp. 761-771.- [A-273]
- [Deriche87] Deriche R. **1987**. Using Canny's criteria to derive a recursively implemented optimal edge detector. *In: International Journal of Computer Vision*, vol. 1, n°2, 1987, pp. 167-187.
- [Cline88] Cline H.E. & al., **1988**. Two algorithms for the three-dimensional reconstruction of tomograms. *In: Medical Physics*, Vol. 15, No 3, May/June 1988, pp. 320-327.
- [Fiorio95] Fiorio, C., **1995**. Approche interpixel en analyse d'images, une topologie et des algorithmes de segmentation. Thèse de doctorat, Université Montpellier II.
- [Fiorio96] Fiorio, C., Gustedt J., **1996**. Two linear union-find strategies for image processing. *In: Theoretical Computer Science A*, 154:165-181.
- [Grace94] Grace L.A., **1994**. Design and evaluation of an optical scanner based log grading and sorting system for Scots Pine (*Pinus Sylvestris* L.) sawlogs. The Swedish University of Agricultural Sciences, Department of forest Products, ISBN 91-576-4848-4, 30 pp.
- [Grönlund96] Grönlund A., Grunberg S., Grönlund U., **1996**. The Swedish stem bank. A unique data base for different silvicultural and wood properties. *Proceedings IUFRO workshop WP S5.01-04*, Hook Sweden June 1994, 71-77.
- [Grundberg91] Grundberg, S., Grönlund, A. **1991**. Methods for reducing data when scanning for internal log defects. *Proceedings of the 4th International Conference on Scanning Technology in the Wood Industry*, San Francisco, USA.
- [Grundberg92] Grundberg, S., Grönlund A. **1992**. Log scanning - extraction of knot geometry in CT- volumes. *Seminar/Workshop on Scanning Technology and Image Processing on Wood*, Skellefteå.
- [Grundberg96] Grundberg, S., Grönlund, A. **1996**. Modelling of the quality in individual logs. *Second IUFRO Workshop WP S 5.01-04. "Connection between silviculture and wood quality through modelling approaches and simulation software"* South Africa August 26-31.
- [Grundberg97] Grundberg S., Grönlund A., **1997**. Simulated grading of logs with an X-ray log scanner - Grading accuracy compared with manual grading. *Scand. J. For. Res.* 12:70-76.
- [Grundberg98] Grundberg, S., Grönlund, A. **1998**. Feature Extraction with the Aid of an X-ray Log Scanner. *The 3rd International Seminar/Workshop on Scanning Technology and Image Processing on Wood*. Skellefteå, Sweden Aug 17-19.

- [Hagman96] Hagman O., 1996. On reflexions of wood. Doctoral thesis. Lulea University of Technology, Division of Wood Technology. ISSN 0348-8373.
- [Hagman99] Hagman O. 1999. Woodmetrics- Imaging devices and processes in wood inspection at Luleå University of Technology. Industrial Lasers & Inspection Conference on Polarization and Colour Techniques in Industrial Inspection in Munich, Germany from 17-18 June.
- [Jaeger93] Jaeger M., Briand D., Borianne Ph., Bonnel F. 1993. Knee anatomy 3D reconstruction and visualization from CT scans. *In*: 2nd European Congress of the European Association of Clinical Anatomy, Munich (Germany), 5-7 sept. 1993, p. 231.
- [Joyeux96] Joyeux H., Jaeger M., Borianne P., Masson B. 1996. Application de l'infographie tridimensionnelle en cancérologie. *In*: Bulletin de l'Académie Nationale de Médecine, 180, No 6, p. 1455-1466, Juillet.1996.
- [Leban99] Leban, J.M., Jaeger, M., 1999. Assessment of timber quality in the forest resources by the mean of tree growth and conversion models. A challenge for today. *In* : COST E10, "Wood Properties for Industrial Use", 13-15 June, Tapada Nacional de Mafra, 11 p.
- [Mongeau93] Mongeau J-P., Beauregard R., Harless T.E.G., 1993, Softwood log shape modelling with shadow scanners. *Wood and fiber science* 25(3):261-277.
- [Oja97] Oja J., 1997. Measuring knots and resin pockets in CT-images of Norway spruce. Licentiate thesis. Lulea University of Technology, Division of Wood Technology. ISSN 1402-1757.
- [Prudhomme98] Prudhomme M., Gaubert-Cristol R., Jaeger M., de Reffye P., Godlewski G. 1998. A new method of three dimensional computer assisted reconstruction of the developing bilary tract. *In* : Surgical and Radiology Anatomy, in press.
- [Saint-André98] Saint-Andre, L. 1998. Modélisation tridimensionnelle des profils de largeur de cerne dans un billon d'Épicéa (*Picea abies* Karst) compte tenu de la mesure de son enveloppe externe et des caractéristiques dendrométriques usuelles de l'arbre d'origine. Thèse de Doctorat de l'ENGREF.. Publication Equipe de Recherches sur la Qualité des Bois 1998, 215 pp. (In French with English summary)
- [Treil95] Treil J., Casteigt J., Roch P., Jaeger M., Cavezian R., Pasquet G. 1995. A new method for 3D cephalometry. *In*: Abstract off XIIth international symposium on morphological sciences, Thessaloniki, Grece, septembre 1995, p. 215-216.
- [Wang97] Wang, X., Hagman, O. and Grundberg, S. 1997. Sorting pulpwood by X-ray scanning. International Mechanical Pulping Conference. Stockholm, Sweden, pp. 395-399, June 9-13.

The Ellipsoidal Skeleton in Medical Applications

F. Banégas
CIRAD*
fbanegas@wanadoo.fr

M. Jaeger
CIRAD
jaeger@cirad.fr

D. Michelucci
EMSE†
micheluc@emse.fr

M. Roelens
EMSE
roelens@emse.fr

Abstract

Rough 3D data images obtained by computed tomography or magnetic resonance imagery are inadequate: this paper proposes a high-level data structure called ellipsoidal skeleton. It is based on a tree of best partitions of the points set and features data compression, multi-level representation capabilities, surface reconstruction, interactive visualization, relevant parameters extraction, automatic matching and recognition.

Keywords: Medical Applications, Dynamic Clustering, Surface Reconstruction, Shape Matching and Recognition, Signature, Tabu Search

1 Introduction

Medical Imaging has to cope with increasingly huge 3D data images: CT-scan or MRI images. The data size does not permit interactive visualization, and lack of structure prevents extraction of relevant informations. Some paradigm is dramatically required, to visualize interactively relevant data with various levels of details, to reconstruct the surface of organs, to extract relevant features for diagnosis or surgical interventions, and to automatically match objects extracted from different 3D data images.

The heart of the solution proposed here is a method to find a tree of best partitions of the set of 3D points from segmented image data. For a given number of classes, the best partition is the one which maximizes the homogeneity of classes and the differences between classes. The partitions in $1, 2 \dots k$ classes are organized in a tree: the E-skeleton, the construction of which is presented in section 2.

For a given level of details, *i.e.* for a given number k of classes, the k -partition permits to reconstruct the surface of the 3D object. Each class is geometrically approximated by an ellipsoid, and all k ellipsoids are merged with a Blinn surface. Possibly, a Barr deformation can be applied to each ellipsoid to best fit the surface. See section 3.

*CIRAD: Centre International de Recherche en Agronomie et Développement, avenue d'Agropolis, BP5035, 34032 Montpellier cedex 1, France.

†EMSE: Ecole des Mines de St-Etienne, 158 cours Fauriel, 42023 St Etienne cedex 2, France.

Due to the hierarchical nature of the E-skeleton, it is then possible to reconstruct and visualize the surface with several levels of detail, as explained in section 4.

The E-skeleton is invariant through isometries, and very robust against discretisation noise, over- and under- sampling, local deformations. Thus two points sets of similar objects give similar E-skeletons. This steadiness property is exploited for automatic matching between shapes, computing distances between shapes, shape recognition: see section 5.

Beyond ellipsoids, it is possible to extract from E-skeleton other parameters relevant for a given application. Examples are given in section 6.

2 Construction of the E-Skeleton

2.1 Expectations and Variances

Let $\mathcal{C} = \text{cloud}(\Omega, P)$ be a cloud of n points, with weights $\Omega \in \mathbb{R}^n$ and coordinates $P = (X \in \mathbb{R}^n, Y \in \mathbb{R}^n, Z \in \mathbb{R}^n)$. They are equally distributed in the interior of the studied organ. In our applications, all initial points have equal weights, but unequal weights occur during computations; moreover some segmentation methods may weight points. The X expectation (or mean value, or center of gravity): $E(\Omega, X)$ or $E(X)$ for short, is:

$$E(X) \stackrel{\text{def}}{=} \frac{\sum_1^n \Omega[i] \times X[i]}{\sum_1^n \Omega[i]}$$

and Y - and Z -expectation are defined similarly. The covariance of X and Y : $C(\Omega, X, Y)$ or $C(X, Y)$ for short, is:

$$\begin{aligned} C(X, Y) &\stackrel{\text{def}}{=} E(XY) - E(X) \times E(Y) \\ &= C(Y, X) \end{aligned}$$

where XY stands for $[X[1] \times Y[1], \dots, X[n] \times Y[n]]$. The variance of X : $V(\Omega, X)$ or $V(X)$ for short, is:

$$\begin{aligned} V(X) &\stackrel{\text{def}}{=} C(X, X) \\ &= E(X^2) - (E(X))^2 \end{aligned}$$

It is also the expectation of the squares of the differences between X and $E(X)$:

$$V(X) = E(X - E(X))^2$$

Some simple and useful properties ($k \in \mathbb{R}$, and $k + X$ is a shortcut for $[k \ k \dots k] + X$):

$$\begin{aligned} E(k + X) &= k + E(X) \\ E(kX) &= k \times E(X) \\ E(X + Y) &= E(X) + E(Y) \\ V(k + X) &= V(X) \\ V(kX) &= k^2 \times V(X) \end{aligned}$$

$$\begin{aligned}
V(X + Y) &= V(X) + V(Y) + 2C(X, Y) \\
C(X, Y + k) &= C(X, Y) \\
C(X, kY) &= k \times C(X, Y) \\
C(X, Y + Z) &= C(X, Y) + C(X, Z)
\end{aligned}$$

The euclidean variance of \mathcal{C} is:

$$V(\mathcal{C}) \stackrel{\text{def}}{=} V(X) + V(Y) + V(Z)$$

Actually it is also the expectation of the squares of the distances between points P and their gravity center ($\bar{X} = E(X)$, $\bar{Y} = E(Y)$, $\bar{Z} = E(Z)$):

$$\begin{aligned}
&E((X - \bar{X})^2 + (Y - \bar{Y})^2 + (Z - \bar{Z})^2) \\
&= E((X - \bar{X})^2) + E((Y - \bar{Y})^2) + E((Z - \bar{Z})^2) \\
&= V(X) + V(Y) + V(Z) = V(\mathcal{C})
\end{aligned}$$

thus, since distances are invariant through isometries (translations, rotations, symmetries and their composition), $V(\mathcal{C})$ as well.

2.2 The Covariance Matrix

The covariance matrix M of the cloud \mathcal{C} is:

$$M \stackrel{\text{def}}{=} \begin{pmatrix} V(X) & C(X, Y) & C(X, Z) \\ C(Y, X) & V(Y) & C(Y, Z) \\ C(Z, X) & C(Z, Y) & V(Z) \end{pmatrix}$$

This matrix is symmetric and positive, so eigenvalues are real and non-negative. In the generic case, the matrix is definite: degeneracies occur for instance when all points in the cloud lie on a plane or a line. Such degeneracies are easy to detect because one or two eigenvalues are much smaller than the greatest one. From now on, only the generic case is considered. Thus the 3 eigenvalues are real and positive: $\lambda_1 \geq \lambda_2 \geq \lambda_3 > 0$.

The 3 corresponding eigenvectors, once normalized, define an orthonormal coordinates system, and are called the main axis of the cloud \mathcal{C} . The major main axis corresponds to λ_1 .

When expressed in this coordinates system (called the natural coordinates system of the cloud), the covariance matrix of the cloud is the diagonal matrix $(\lambda_1, \lambda_2, \lambda_3)$, i.e. $C(X, Y) = C(Y, Z) = C(X, Z) = 0$. Up to orientations, this natural coordinates system is unique. Accounting for orientations, there are 2 such coordinates systems in one dimension, and 2^{d-1} in d dimensions when $d > 1$.



Figure 1: There are 2 natural coordinates systems in 2D.

The covariance matrix has same eigenvectors as the matrix of inertia, more familiar to mechanical engineers, and which has a strong physical meaning.

2.3 Best Partitions

Let $\{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ be a k -partition (i.e. a partition in k non empty classes) of the cloud \mathcal{C} . The euclidean variance $V(\mathcal{C})$ of the cloud \mathcal{C} is decomposed into two parts: the sum of all euclidean variances of clouds $\mathcal{C}_1, \dots, \mathcal{C}_k$ (the latter are called: intraclass), and the interclass variance $V(\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k\})$:

$$V(\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k\}) \stackrel{\text{def}}{=} V(\mathcal{C}) - \sum_{i=1}^k V(\mathcal{C}_i)$$

Let $\Omega_i, P_i = (X_i, Y_i, Z_i)$ be the vectors of weights and coordinates of the cloud \mathcal{C}_i . Let n_i be the number of points in \mathcal{C}_i . $G_i = (E(\Omega_i, X_i), E(\Omega_i, Y_i), E(\Omega_i, Z_i))$ is the gravity center of \mathcal{C}_i , and $w_i = \sum_{j=1}^{n_i} \Omega_i[j]$ is the weight of G_i , i.e. the weights sum of points in \mathcal{C}_i . Pose $G = (G_1, G_2, \dots, G_k)$ and $W = (w_1, w_2, \dots, w_k)$: they define a cloud $\mathcal{G} = \text{cloud}(W, G)$, whose euclidean variance is the euclidean interclass variance of the partition $\{\mathcal{C}_1, \dots, \mathcal{C}_k\}$.

The best k -partition is the one with maximal interclass variance, and thus with minimal sum of intraclass variances, since the sum of both is constant. It means classes are as different and homogeneous as possible.

Extreme cases: if $k = 1$, $V(\{\mathcal{C}_1\}) = 0$: the interclass variance vanishes. If $k = n$ (there is one class per point), $V(\mathcal{C}) = V(\{\mathcal{C}_1, \dots, \mathcal{C}_n\})$ and $V(\mathcal{C}_i) = 0, i = 1 \dots n$.

The progressive dynamic clustering algorithm finds the best partitions.

2.4 Computing Best Partitions

2.4.1 The Dynamic Clustering Algorithm

To compute the best k -partition of a cloud, i.e. the partition that maximizes the interclass variance, the DC algorithm [15] proceeds as follow:

Choose an initial k -partition

Repeat

For each class $\mathcal{C}_i, i = 1 \dots k$

Compute G_i the center of gravity of \mathcal{C}_i

Empty the class \mathcal{C}_i

EndForEach

For each point p_j

Add p_j to class \mathcal{C}_i where G_i is closest to p_j

EndForEach

Until neither of G_i change

This algorithm is guaranteed to find the optimal solution; oscillations due to degeneracies or inaccuracies may happen in some case but are easily avoided. The DC algorithm has two main drawbacks: the number of steps can be very important, and the initial choice of the partition has a great influence on this number. It is therefore crucial to start with a good initial partition.

2.4.2 The Progressive DCA

To improve the DC algorithm, we construct a progressive DCA: initially, it is trivial to find the optimal 1-partition. Assume we know the optimal k -partition $(\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k)$. For each and every class $\mathcal{C}_i, i = 1 \dots k$, we split \mathcal{C}_i by the plane containing G_i and orthogonal to the major main axis of \mathcal{C}_i ; a DCA is used to split optimally \mathcal{C}_i in two sub-classes \mathcal{C}_i' and \mathcal{C}_i'' ; we then compute the ratio $r_i \stackrel{\text{def}}{=} \frac{V(\mathcal{C}_i') + V(\mathcal{C}_i'')}{V(\mathcal{C}_i)}$.

The initial $k + 1$ -partition for the DCA: $\{\mathcal{C}_1, \dots, \mathcal{C}_i', \mathcal{C}_i'', \dots, \mathcal{C}_k\}$ is obtained by splitting the class \mathcal{C}_i with smallest r_i ratio.

2.4.3 Optimal Maximal Number of Classes

The simplest criterion to stop the subdivision process is the ratio of the sum of intraclass variance to the cloud variance: when this ratio becomes less than a given threshold, typically 5%, the subdivision is stopped. This ratio is a monotonic decreasing function of k .

Many methods have been proposed to find optimal values for k , but they are rather complex to implement and are often application specific [33, 21]. For our application, the simplest criterion is sufficient.

2.4.4 An Improvement

An easy way to speed up the progressive DCA is to undersample the cloud \mathcal{C} into a cloud \mathcal{C}' , keeping for instance only the central voxel (or the gravity center) of each cube of $3 \times 3 \times 3$ contiguous voxels in the 3D image of \mathcal{C} , to compute the best k -partition P' of \mathcal{C}' , to replace in P' the voxels of \mathcal{C}' by the corresponding voxels in \mathcal{C} , and to use the latter as an initial k -partition for the DCA—maybe this principle can be used recursively, while the cloud has enough points, say k^2 , but this last recursive optimization has not been tried.

This optimization uses the grid structure of 3D data images, which is always present in medical imaging. We have not investigated the use of octree data structures to speed up the DCA.

2.4.5 Examples

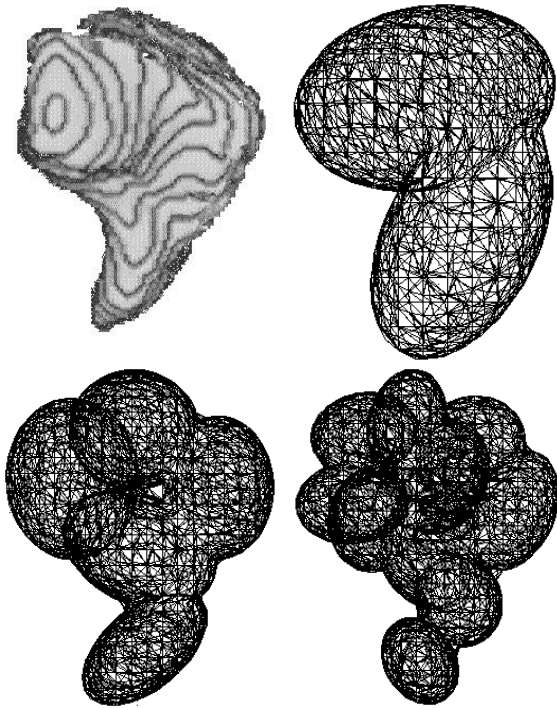


Figure 2: A carpal bone, the hamatum: the cloud, 2-, 5-, and 10-partition.

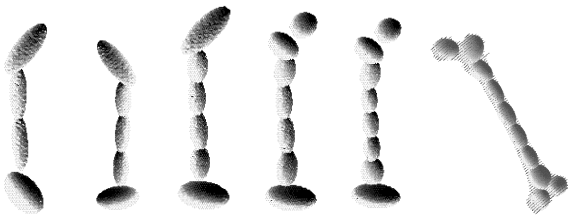


Figure 3: A femur. 4-, 5-, 6-, 7-, 8-, 10- partitions.

2.4.6 Mahalanobis Variant

With increasing k , classes of k -partitions become more and more spherical, so that a tubular part (for instance the middle part of a femur) is partitioned by a string of balls. One may prefer to keep such a part as a single class. A solution is to apply to the initial cloud \mathcal{C} , with eigenvalues λ_i , a scaling transform, with factors $(\lambda_1^{-1/2}, \lambda_2^{-1/2}, \lambda_3^{-1/2})$, in the natural coordinates system of \mathcal{C} . Let \mathcal{C}' be the resulting cloud: all eigenvalues of its covariance matrix are equal to 1. Then the k -partition of \mathcal{C}' is computed with the progressive DCA (with or without optimizations). Due to the 1-1 correspondance between \mathcal{C} and \mathcal{C}' , the k -partition of \mathcal{C}' is also a k -partition of \mathcal{C} , which can be called the optimal *normalized* k -partition of \mathcal{C} . This variant is equivalent to using Mahalanobis distance. An advantage of this approach is that two clouds equal up to some affine transform have equal optimal normalized k -partitions, by definition. This variant has been tested, but unfortunately it gives counter intuitive results with increasing k (see [3] for details), and we do not retain it.

2.5 Construction of the Tree

A tree: the E-skeleton, keeps track of the subdivision process. Each node or leaf represents a class. At level k (1 for the root) this tree has k nodes, which represent the optimal k -partition. At each level k , except the leaves level, $k - 1$ nodes have exactly one children, and one node has two child: the latter represents the class which was split to obtain the optimal $k + 1$ -partition.

Apart its childhood, each node stores its gravity center, its weight, its number of points, its covariance matrix, its eigenvalues and eigenvectors (the main axis). Geometric values α and modal deformation u , which are used for geometric reconstruction and defined below, are also stored.

3 Reconstruction of the Surface

3.1 Previous Works

Various approaches have been proposed for surface reconstruction, from tetrahedral reconstruction [9] to generalized cylinders [35]. Such techniques remain locally based or need strong topological information about contour points, which can be ambiguous in some case.

Methods that are more generic exist, like active contours [11] or snakes, offering compact analytical expression of 3D objects without the need of any knowledge of the topology. The concept of reconstruction with multiple primitives [24] can be seen as a generalization of these snakes, and has become very popular [13, 38, 6]. Formalisms have been proposed to provide strong bases for primitive combining [2, 31, 39, 28] and fitting [4, 37].

Parametric primitives [14], while providing great flexibility, are often complex to combine. Implicit surfaces [7] simplify this issue but must be wisely used to avoid the evaluation of numerous expensive equation. It is why implicit primitives must not only be positioned carefully according to the point cloud, but also be as representative as possible [1]. They should offer a level of realism of their own.

Combined with this feature, level of detail generation must be present: multi-resolution representation [29] can be achieved via triangle decimation [30], progressive meshes [20] or discretization step control [22].

3.2 Our Approach

An optimal k -partition of the cloud can be used to construct a geometric approximation of the surface of the object. To get a rough

approximation, sufficient when the object is far from the eye, we approximate each class with its best ellipsoid (section 3.3), and union them. To get a smoother and more accurate approximation, we use the Blinn formalism [7], inspired from molecular chemistry: a surface is defined as the set of points where a potential is equal to 1. This potential is defined as the sum of elementary potentials: in our model, each class \mathcal{C}_i of the skeleton is attached a geometric primitive, which emits an elementary potential $P_i(x, y, z)$ in all space:

$$P_i(x, y, z) \stackrel{\text{def}}{=} e^{-\alpha_i d_i(x, y, z)} \quad (1)$$

where $d_i(x, y, z)$ is some distance to the geometric primitive boundary, and $\alpha > 0$ some adjusting parameter (the slope of the potential curve as a function of the distance d_i). The distance $d_i(x, y, z)$ must be zero on the boundary of the geometric primitive of the class, negative inside the geometric primitive, and positive outside. We now present possible geometric primitives for a class, and the corresponding distance.

3.3 The Best Ellipsoid of a Class

The geometric primitive for a class can be an ellipsoid: the center is the gravity center $G = (x_g, y_g, z_g)$ of the class, its main axis are the main axis of the class (i.e. eigenvectors of its covariance matrix), and its radius a, b, c are $\gamma\sqrt{\lambda_i}$, $i = 1, 2, 3$, where λ_i are eigenvalues of the related covariance matrix, and γ is a scaling constant.

There are 2 possible choices for γ . Either we want that the best ellipsoid is the cloud itself when the latter is a regularly sampled ellipsoid: then $\gamma = \sqrt{d+2}$, where d is the space dimension; in our case, $d = 3 \Rightarrow \gamma = \sqrt{5} \approx 2.236068$. Or we use the big numbers law: if the random variable X follows a normal law, the interval $[-2\sqrt{V(X)}, 2\sqrt{V(X)}]$ contains 90% of the samples, and so $\gamma = 2$, in all dimensions. We choose the latter.

In the natural coordinates system of the class, the ellipsoid has equation: $E(x, y, z) = 0$ where:

$$E(x, y, z) = \frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} - 1$$

Expressing $E(x, y, z)$ in world coordinates gives the simplest possible distance for equation 1. More precisely, this equation becomes:

$$E(p) = (p - G)Q(p - G)^t$$

where G is the gravity center, $Q = RDR^t$, $D = 1/\gamma^2 \times \text{diag}(\lambda_1^{-1/2}, \lambda_2^{-1/2}, \lambda_3^{-1/2})$, and R is the matrix of eigenvectors of the covariance matrix, so that $R^t = R^{-1}$ and $M = R^t \text{diag}(\lambda_1, \lambda_2, \lambda_3)R$. The diagonalization of M can be computed with Jacobi rotations [27].

3.4 The Best Superquadric of a Class

Superquadrics are another geometric primitive. In the natural coordinates system of the class, they have equation $S(x, y, z) = 0$ with:

$$S(x, y, z) = \left(\left(\frac{x}{a} \right)^{\frac{2}{\epsilon_1}} + \left(\frac{y}{b} \right)^{\frac{2}{\epsilon_1}} \right)^{\frac{\epsilon_1}{\epsilon_2}} + \left(\frac{z}{c} \right)^{\frac{2}{\epsilon_2}} - 1$$

with $\epsilon_1 > 0, \epsilon_2 > 0$. For $\epsilon_1 = \epsilon_2 = 1$, superquadrics reduce to ellipsoids. Moreover, these parameters allow a continuous characterization of the shape of each class, adding useful information for pattern matching 5.2.

Expressing $S(x, y, z)$ in world coordinates gives the corresponding distance function in equation 1. To find best parameters ϵ_1, ϵ_2 , some optimization is required (section 3.6).

3.5 Deformed Superquadric of a Class

Deformed superquadrics are the last possible geometric primitive of our model. We use the modal deformation D_u , as defined by Barr [4]. u is a 30 dimensional vector: $(t_x t_y t_z \phi \theta \psi a b c \mu_1 \dots \mu_p)$ whose coordinates describe elementary deformations (translation, rotation ... bending, twisting). u defines an affine map $D_u(p)$ in every point p of space, i.e. a 4×4 matrix in homogeneous coordinates, which is a function of $p \in \mathbb{R}^3$. If $S(p) = 0$ is the equation of the non deformed superquadric, then the deformed superquadric has equation: $S_u(p) = S(p \times D_u(p)^{-1}) = 0$. Expressing S_u in world coordinates gives the distance function for the deformed superquadric in equation 1. Some optimization is required (section 3.6) to find the best vector u .

3.6 Optimisation

In the simplest case the k best ellipsoids are just merged with $\alpha_i = 1$, or even just unioned: no optimization is needed. It gives a schematic representation which is perfect for pedagogical illustrations.

If a more accurate reconstruction is needed, we have first to find the best geometric primitive for each sub-cloud, second to find best α_i for merging the k best geometric primitives. These two stages simplify the resolution process. We use the same resolution method for both problems, so we discuss only the first one.

The optimization method has to find the best values for parameters $\epsilon_1, \epsilon_2, u$, in order for the deformed superquadric to best fit the boundary points of the class. The optimization minimizes the error measure: $e = \sum_{p \in B} d(p)^2 / \text{card}(B)$ where B is the set of boundary points, $\text{card}(B)$ its cardinality, and $d(p)$ is the distance function the parameters of which are optimized. B may be undersampled, to speed up. A point of the cloud is a boundary point iff one of its 6 neighbors, in the initial 3D data image, does not belong to the same cloud (It is the second place where the E-skeleton software uses the grid structure of 3D data images). Several optimization methods have been tried:

Descent methods (Levenberg-Marquardt, conjugate gradients) often stall in local minima.

Homotopy projects boundary points on the undeformed geometric primitive, moves them incrementally to their right location, and at each step performs a descent method. It works much better, but unfortunately not always, and it is time consuming.

Tabu, a stochastic method, resembles simulated annealing [18, 19]. Tabu walks in a discretized search space: at a current vertex of this search space, it randomly chooses t_n vertices in some neighborhood, eliminates those which were recently visited (tabu manages a tabu list of the last t_v visited points, to avoid cycling), and jumps to the best vertex, even if it is worse than the current one or the best vertex met so far. The neighborhood radius decreases during the walk. Tabu stops when it has found a solution with error less than a prescribed threshold, or after a prescribed number of steps. For our application, values $t_n = 5$ and $t_v = 10$ were sufficient to find the optimal solution. Tabu is time consuming, but easy to implement and the most robust method so far. Its solution is then polished with a classic descent method.

3.7 Reconstruction Examples

After optimization, the quality of the solution is measured with the more accurate, but more time consuming, error measure $e_{me} = \sum_{p \in B} \delta(p) / \text{card}(B)$ or $e_{max} = \max_{p \in B} \delta(p)$ where [36]:

$$\delta(p) = \frac{|d(p)|}{\|\nabla d(p)\|} \approx \text{dist}(p, B)$$

The mean error is typically divided by 10, and the maximal one by 3 or 4, relatively to the simplest approximation which blends all best ellipsoids with $\alpha_i = 1$.

The table 1 gives some results for two objects: a femur and a hamatum. The following parameters are given: number of primitives n_p , number of boundary points B_p , maximal length l of the cloud (in millimeters), mean error e_{me} and maximal error e_{max} (in millimeters), and computation time t (in minutes), on a SGI O^2 workstation, R5000 at 180 Mhz.

A low number of geometric primitives gives visually satisfying results. A better accuracy may be achieved, but at the cost of a bigger number of classes, and is irrelevant for medical application, where the 3D data generation process (acquisition and segmentation) may introduce errors about 7 millimeters for a femur. Another argument against a greater accuracy is the plasticity of some organs (lever, heart, etc).

Table 1: Performance results

Object	B_p	n_p	l	e_{me}	e_{max}	t
Hamatum	7500	4	20	1.1	3.2	4'
Hamatum	7500	5	20	0.8	2.8	5'
Femur	25000	6	400	2.4	5	20'
Femur	25000	8	400	2	4.7	15'

It might be noticed that the computation time for the femur is shorter for 8 primitives than for 6: this is due to the fact that ellipsoids for 8 primitives are more spherical, that speeds up optimization.

Figures 4 to 7 show some reconstruction examples. Surfaces are tessellated with some variant of the Marching Cubes algorithm [22] (see section 4). The computation of the E-skeleton and the surface reconstruction (without marching cubes) needs typically from 2 minutes for the hamatum with 38,767 points with $k = 10$, to 6 minutes for the femur with 152,727 points also with $k = 10$ (times for Intel Pentium II at 233 MHz). The E-skeleton, including the analytical description of the surface (u vectors, α_i), is about one thousand more compact than the initial 3D data image. It is also about one hundred more compact than a medium-quality polygonization of the initial 3D image.

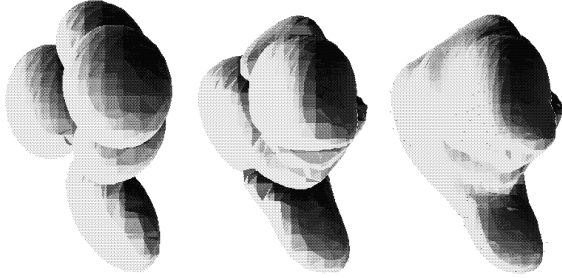


Figure 4: Reconstruction of hamatum: ellipsoids, deformed ellipsoids, blended deformed ellipsoids.

3.8 Implicit Surface with Even Degree

This paragraph describes another solution, not yet implemented: the geometric primitive for approximating a class can be any implicit algebraic surface with even degree $2d$ (to ensure bounded-



Figure 5: Deformation of ellipsoids for the 4-partition of the hamatum.



Figure 6: Reconstruction of the eight carpal bones: each bone is separately blended.

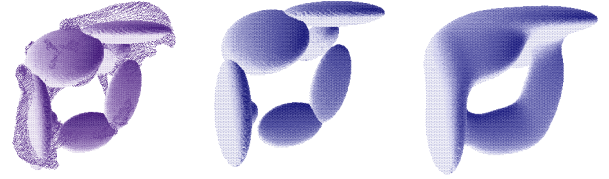


Figure 7: Reconstruction of an iliac bone: cloud and ellipsoids, ellipsoids, merged undeformed ellipsoids.

ness), having equation $F(x, y, z) = 0$ where:

$$F(x, y, z) = \sum_{0 \leq i+j+k \leq 2d} c_{i,j,k} x^i y^j z^k \quad (2)$$

Moreover, it is convenient to impose $c_{0,0,0} = 1$ (which implies the origin does not lie on the surface: some translation can be applied to the cloud to ensure that). For $p = (p_x, p_y, p_z)$ a given point, $F(p)$ is then a computable linear expression in $C = [c_{i,j,k}]$: $F(p) = 1 + C.l$, where l is the vector $[p_x^i \times p_y^j \times p_z^k]$ (in the same order than the $c_{i,j,k}$ in the C vector); if p is a boundary point of the class, we want $F(p) = 1 + C.l$ as small as possible. Expressing this constraint for all boundary points p , we have to minimize the euclidean norm of the vector $[1, \dots, 1] + C.L$. It is a least square problem, whose analytical solution is $C = [-1, \dots, -1]L^+$, where L^+ is the pseudo inverse of L : $L^+ = L^t(LL^t)^{-1}$. An advantage of this formulation is that fast least square methods can be used [27].

Finally, the power basis: $(1, x, x^2 \dots) \times (1, y, y^2 \dots) \times (1, z, z^2 \dots)$ was used only for simplicity in formula 2: as well known, it has not the property of affine invariance. The Bernstein

basis does have, and should be preferred in order to avoid the shape of the resulting surface to depend on the used coordinates system.

4 Multi Level Reconstruction and Visualization

At each level, the E-skeleton stores for an object an analytical form: the surface is defined by an equation $F(x, y, z) = 1$. For interactive applications, a polygonal definition of the object is needed (this form is required by advanced graphic workstations): a meshing algorithm is then used. We choose a tetrahedric variant of the Marching Cubes algorithm [22]: this algorithm may use an extra parameter, the discretization step s .

Each mesh generation needs less than one second: we don't have to evaluate the potential in each vertex of the Marching Cube grid. Instead, for each ellipsoid, we start from its center, walk upwards until we reach the surface boundary. Either the corresponding voxel already contains some reference to a polygon: then this connected component of the surface has already been tessellated and we are done; or the corresponding voxel is empty, and we start a new meshing stage from this voxel, marking the visited voxels. Using continuity avoids computing potentials in all voxels.

We have two parameters to choose a polygonal mesh: the semantic level k and the discretization step s . This provides a great flexibility for surface rendering. To control the number of meshes, k and s are decreasing staircase functions of the distance between the eye and the object. Instead of precomputing all meshes at all levels and all discretization steps, meshes are lazily computed (*i.e.* when needed) and saved to disk after creation.

Two extra parameters may also be used to monitor the surface reconstruction process: either we use surface blending or boolean union, either we deform geometric primitives or not. For union of undeformed ellipsoids, it should be interesting to polygonize directly: the marching cube algorithm is no more needed.

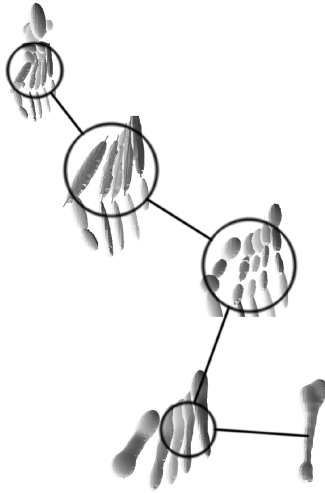


Figure 8: Dynamic rendering for visualization and exploration on tarsal bones.

5 Shape Recognition

5.1 Previous Works

Many techniques have been investigated, that may be divided into two classes: either they rely on external [8, 1, 35] or internal ex-

pression of objects. Former ones are based on the surface definition of the object (B-rep, polygonal ou B-spline representation), and latter ones use area descriptor methods (medial axis, shape decomposition). Such descriptors are either structural, like medial axis [16, 25] or scalar [32].

Shape decomposition is a powerful technique as it achieves data reduction by imposing a particular organization on it [24, 26]. Pixel- or voxel-scaled information is organized into larger models, which provides multiscale capabilities [23]. Some theories even suggest that such decomposition is present in human perception [5].

The model we propose in this paper combines all aspects of internal description-based algorithms, as it uses both structural (*i.e.* geometric) and scalar (*i.e.* by extracting relevant parameters or signatures) descriptions in a hierarchical fashion thanks to semantically zoomable shape decomposition approach.

5.2 Matching k -Partitions

This section presents the automatic comparison of two objects, at a given level k of detail, *i.e.* both clouds are described by their best k -partition. When the size is not relevant (*i.e.* we want to find equal two homothetic objects), some homothecy is applied to give them volume 1. We assume both clouds are expressed in their natural coordinates systems (expectations, variances, etc are updated accordingly).

5.2.1 Class Signature and Distance

To compute a *distance between two classes*, we attach a characteristic vector, or *signature* [34], to each class, then compute their Manhattan distance (the euclidean distance is another possibility). Our software associates to a class \mathcal{C}_j the signature vector: $(\sqrt{\lambda_{j1}}, \sqrt{\lambda_{j2}}, \sqrt{\lambda_{j3}}, \rho_j)$ where λ_{ji} are eigenvalues of the covariance matrix of \mathcal{C}_j , and ρ_j is the distance between the gravity center of \mathcal{C}_j and the one of the whole cloud \mathcal{C} (*i.e.* the origin). Other choices are possible, which for example take into account angles, like angles cosines between main axis of \mathcal{C}_j and main axis of the whole cloud \mathcal{C} .

Note it makes no sense to directly compare coordinates in \mathcal{C} and coordinates in \mathcal{C}' , due to the existence of several natural coordinates systems.

In our applications, the simplest definition was always sufficient to match objects. More on chirality and signatures in sections 5.5, 5.6.

5.2.2 The Matching Problem

Let \mathcal{C} and \mathcal{C}' be two clouds described by two k -partitions: $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2 \dots \cup \mathcal{C}_k$ and $\mathcal{C}' = \mathcal{C}'_1 \cup \mathcal{C}'_2 \dots \cup \mathcal{C}'_k$. The distance between \mathcal{C} and \mathcal{C}' is:

$$\text{dist}(\mathcal{C}, \mathcal{C}') = \min \text{dist}(\mathcal{C}_1, \sigma(\mathcal{C}'_1)) + \dots + \text{dist}(\mathcal{C}_k, \sigma(\mathcal{C}'_k))$$

where σ describes the $k!$ bijections from $\{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ to $\{\mathcal{C}'_1, \dots, \mathcal{C}'_k\}$. This is a minimal matching problem: given the square $k \times k$ matrix of all distances between \mathcal{C}_i and \mathcal{C}'_j , find k entries, one per line, one per column, with minimal sum. It is also a max flow min cost problem: each class of \mathcal{C} and \mathcal{C}' is represented by a vertex; all vertices of \mathcal{C} are linked to all vertices of \mathcal{C}' , each edge $\mathcal{C}_i \mathcal{C}'_j$ has cost $\text{dist}(\mathcal{C}_i, \mathcal{C}'_j)$. A source vertex A is connected to all vertices of \mathcal{C} , with cost zero, and all vertices of \mathcal{C}' are connected to a sink vertex Z also with cost zero. All edges have minimal capacity 0, maximal capacity 1. The problem is to find the maximal flow from A to Z , at minimal cost. This is a well known problem, and many algorithms exist. The optimal ones need $O(k^2 + \log k)$ time [17, 12].

5.3 Recognition of an Object

Recognition of a given shape \mathcal{C} is finding the closest shape \mathcal{S} in a given library of shapes $\mathcal{S}_1, \dots, \mathcal{S}_N$, and finding the differences between \mathcal{C} and \mathcal{S} .

Our approach describes all shapes by their E-skeletons. To speed up the recognition of the shape \mathcal{C} , \mathcal{C} is compared to each library shape at the first level of detail: only 1-partitions are compared. Some prescribed ratio, typically 90%, of the library shapes, the farthest from \mathcal{C} , are discarded. The remaining library shapes are then compared to \mathcal{C} at the second level of detail, using their 2-partitions. Again, the worse 90% library shapes are eliminated, and so on, until it remains 10 or less library shapes: since each of these steps divides by ten the number of possible library shapes, $O(\log_{10} N)$ steps are needed to find the best library shapes. \mathcal{C} is then compared to all remaining library shapes, with the maximal possible level of detail. The shape with smaller distance is the searched library shape \mathcal{S} .

5.4 Isomorphism

One may think that the E-skeletons of similar objects must be isomorphic. It suggests an alternate definition for recognition, and the following method, which we had no time enough to investigate.

Assume the bijection σ from the classes $C_i, i = 1 \dots k$ of the k -partition of \mathcal{C} to the classes $C'_i, i = 1 \dots k$ of the k -partition of \mathcal{C}' is known: initially it is true for $k = 1$. Let C_j be the split class in \mathcal{C} , C_j has child C'_j and C''_j . Then the split class in \mathcal{C}' must be $\sigma(C_j)$; if $\sigma(C_j)$ is really the split class in \mathcal{C}' , we can extend the isomorphism between the two trees: if $\sigma(C_j)$ has child c', c'' , then there is only two possibilities to extend σ for the $k + 1$ -partition: either $\sigma(C'_j) = c'$ and $\sigma(C''_j) = c''$, or $\sigma(C'_j) = c''$ and $\sigma(C''_j) = c'$. It is a very simple minimum matching problem, for a 2×2 matrix. The deeper two E-skeletons are isomorphic this way, the greater is their similarity.

5.5 About Chirality

A funny feature of our recognition procedure is that it finds equal two enantiomorph objects (*i.e.* differing by a symmetry, like the left and right hand), which makes sense in many medical applications.

It is not due to the existence of several natural coordinates systems. It is due to the fact that only various signatures for classes in the clouds \mathcal{C} and \mathcal{C}' are compared—the class signature $(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \sqrt{\lambda_3}, r)$ in section 5.2.1—and they are invariant by symmetry.

Signatures for k -partitions (and not for classes, like in section 5.2.1) discriminating left and right are possible only if points coordinates of clouds are always expressed in coordinates system with the same orientation (for instance: with your head at $(0, 0, 0)$, your feet at $(0, 0, -1)$, and looking at $(1, 0, 0)$, you have always the point $(0, 1, 0)$ on your left). Assuming that, the simplest signature for a $k \geq 4$ -partition of a cloud \mathcal{C} , discriminating left and right, is as follows: consider 4 classes C_1, \dots, C_4 of \mathcal{C} , with gravity centers G_1, \dots, G_4 , affinely independent (*i.e.* G_1, G_2, G_3, G_4 must be non coplanar, and even "far from coplanar"). Then the determinant $\det(G_1, G_2, G_3, G_4)$, where G_i are expressed in homogeneous coordinates, is $3! = 6$ times the signed volume of the tetrahedron G_1, G_2, G_3, G_4 , and it is the searched signature: it will discriminate left and right, whatever the choice of the natural coordinates systems. After matching this k -partition of \mathcal{C} with the k -partition of \mathcal{C}' , we compute $\det(G_1, G_2, G_3, G_4)$ and $\det(\sigma(G_1), \sigma(G_2), \sigma(G_3), \sigma(G_4))$, where σ is the bijection induced by the minimal matching: it they have the same sign, they have equal orientation, otherwise opposite. Thus it is possible to not confuse left and right, if needed.

5.6 Signatures for k -Partitions

So it is possible to define signatures for k -partitions: for instance, the set of all distances between the k gravity centers of the k -partition, the set of signed (or unsigned if we want to recognize modulo symmetry) volumes of tetrahedra. Once two k -partitions have been matched, it makes sense to compute the distance between two such signatures. Again, we did not need such sophisticated tools.

5.7 Recognition Examples

To test the recognition capability of the E-skeleton, we made a reference library with eight carpal bones. We then compare 4 exams (32 bones) of various patients with library E-skeletons. Table 2 is an excerpt of the distance matrix for hamatum, capitatum and lunatum for the five patients (values have been rounded). It is very easy to see on the matrix that bones have been correctly recognized.

Another test involved upper and lower jaws of an adult: all teeth have been compared, and left and right teeth have been matched correctly (upper and lower teeth are not symmetrical). For one tooth (an incisive), the symmetrical tooth has been identified successfully, but with a greater distance: it was due to a real pathology on the tooth.

The same test was also performed on left and right femur of an adult: due to the chirality independence of the class signature, the two bones were matched.

6 Extraction of Relevant Parameters

From the E-skeleton, it is possible to extract other parameters than those used for the signature. For example, an analysis editor was developed to find correlation between organs (these correlations are called *allometries*), and between organs and various criteria like age, gender or genetic parameters. This editor allowed to measure the well-known correlation between the shape of the carpal bones and the age of the patient.

More precisely, the hamatum may have a protuberance called *hamulus* that appears when bones are matured. The E-skeleton shows this protuberance by the comparison between main axis of the E-skeleton at levels 1 and 2: for a young patient, main axis of the two ellipsoids at level 2 are roughly orthogonal to the main axis of the level one. For an adult, on the contrary, the angle of one of the two ellipsoids (which represents the hamulus) is about 20 degrees.

This editor also reveals that some correlation exists between hamulus maturation and the gender of the patient. Many other criteria were tested and eliminated thanks to the flexibility of the tool. Further details may be read in [10].

In this example, proposals of correlation tests were made by an expert (physician) but it should be possible, if enough data samples were available, to automatically find relevant correlations with data analysis methods, or even artificial intelligence techniques.

7 Conclusion

This model was integrated in a commercial medical application called *Corpus 2000* developed by the MEB team at CIRAD (<http://www.cirad.fr/>) for viewing and analyzing biological entities. This software is available on various platforms (presently SGI IRIX, HP UX and Windows NT4/95/98).

The E-skeleton model presented was used in a tree modeler, which proves that this model may be used in other application domains. The E-skeleton may be considered as a generic model, that combines compacity of the data structures and multi-level representation, interactive manipulation and visualization, and capacities of

Table 2: Distance matrix for hamatum, capitatum and lunatum

	H1	H2	H3	H4	H5	C1	C2	C3	C4	C5	L1	L2	L3	L4	L5
H1	0	8	3	3	3	67	65	64	63	65	54	53	52	53	53
H2	8	0	8	7	7	69	67	66	66	67	54	53	52	54	53
H3	3	8	0	4	4	67	65	64	63	65	53	52	52	52	52
H4	3	7	4	0	1	68	66	65	64	66	53	53	51	53	53
H5	3	7	4	1	0	68	66	65	65	66	53	53	52	53	53
C1	67	69	67	68	68	0	4	5	6	3	68	66	68	68	66
C2	65	67	65	66	66	4	0	6	7	3	66	64	65	65	64
C3	64	66	64	65	65	5	6	0	3	5	68	66	68	67	66
C4	63	66	63	64	65	6	7	3	0	6	67	65	66	66	65
C5	65	67	65	66	66	3	3	5	6	0	67	65	66	66	65
L1	54	54	53	53	53	68	66	68	67	67	0	5	2	3	4
L2	53	53	52	53	53	66	64	66	65	65	5	0	4	3	1
L3	52	52	52	51	52	68	65	68	66	66	2	4	0	2	3
L4	53	54	52	53	53	68	65	67	66	66	3	3	2	0	2
L5	53	53	52	53	53	66	64	66	65	65	4	1	3	2	0

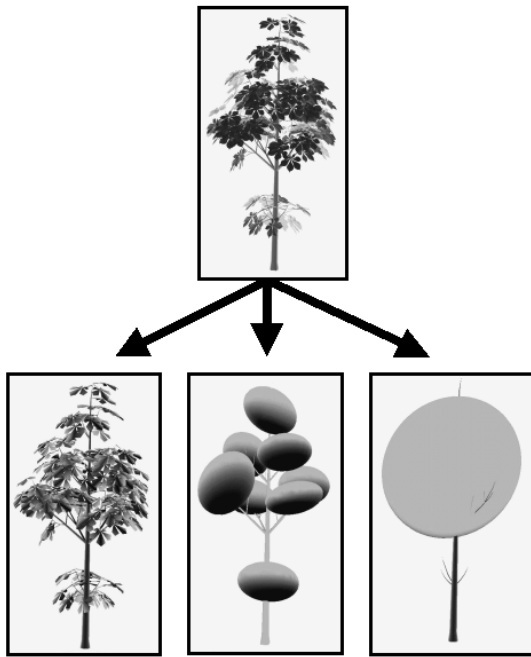


Figure 9: Multi-level representation of a tree.

automatic matching. Depending on the application domain, it also helps experts to find relevant parameters or correlations.

Further developments could involve:

- extension of geometric primitives for surface reconstruction;
- improvement of the E-skeleton by inserting a spring-mass pair for each ellipsoid or primitive, allowing extensions towards dynamic models or physically-based models;
- automatic detection of correlations or allometries inside data;
- automatic allometry-based synthesis of organs;

- construction of an atlas of anatomical shapes at several ages, including shapes with pathologies to be recognized;
- applications in CAD, like feature-based recognition and modelling;
- use of feature recognition to help segmentation of original images.

References

- [1] A. Jaklič, A. Leonardis and F. Solina. Superquadrics for segmenting and modeling range data. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19, pages 1289–1295. IEEE, November 1997.
- [2] A. Sourin, A. Pasko, V. Adzhiev and V. Savchenko. Function representation in geometric modelling : concepts, implementation and applications. In *The Visual Computer*, pages 429–446. Springer-Verlag, 1995.
- [3] F. Banégas. *Caractérisation et reconstruction de solides tridimensionnels par squelette ellipsoïdal*. PhD thesis, Ecole Nationale des Mines de St-Etienne, 158 cours Fauriel, 42023 St Etienne cedex 2, France, 2000.
- [4] A. Barr. Global and local deformations of solid primitives. In *Computer Graphics*, volume 18, pages 21–30. ACM Press, 1984.
- [5] I. Biederman. Recognition-by-components: a theory image human understanding. *Psychological review*, 94:115–147, 1987.
- [6] E. Bittar, N. Tsingos, and M.P. Gascuel. Automatic reconstruction of unstructured 3d data: combining a medial axis and implicit surfaces. In F. Post and M. Gobel, editors, *Eurographics '95*, volume 14, 1995.
- [7] J.F. Blinn. A generalization of algebraic surface drawing. In *ACM Trans. on Graphics*, volume 1, pages 235–256. ACM Press, 1982.

- [8] J. D. Boissonnat. Geometric structures for three-dimensional shape representation. In *ACM Transactions on Graphics*, volume 3, pages 266–286. ACM Press, 1984.
- [9] J. D. Boissonnat. Shape reconstruction from planar cross sections. In *Computer Vision, Graphics and Image Processing*, 44, 1988.
- [10] F. Canovas, F. Banégas, C. Cyteval, M. Jaeger, A. Dimeglio, C. Sultan, and F. Bonnel. Carpal bone maturation assessment by image analysis from ct-scans. *Journal of radiology*, 2000. to be published.
- [11] L.D. Cohen. On active contour models and balloons. *Computer Vision Graphics Image Proceedings*, 53:211–218, 1991.
- [12] T. Cormen, C. Leiserson, and R. Rivest. *Foundations of Computer Science*. MIT Press, Cambridge, Massachusetts, 1990.
- [13] P. Bertolino D. Attali and A. Montanvert. Using polyballs to approximate shapes and skeletons, 1994.
- [14] G. Danuser and M. Stricker. Parametric model fitting : From inlier characterization to outlier detection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 20, pages 263–280. IEEE, March 1998.
- [15] E. Diday. Une nouvelle méthode en classification automatique et reconnaissance des formes : la méthode des nuées dynamiques. In *Rev. Statist. Appl.*, volume 19, pages 19–33, 1971.
- [16] E. Ferley, M.P. Gascuel, and D. Attali. Skeletal reconstruction of branching shapes. In *Implicit Surfaces '96*, Eindhoven (The Netherlands), October 1996.
- [17] Z. Galil. Efficient algorithms for finding maximum matching in graphs. *Computing Surveys*, 18(1), march 1986.
- [18] F. Glover. Tabu search; part I. *ORSA Journal on Computing*, 1(3):190–206, 1989.
- [19] F. Glover. Tabu search; part II. *ORSA Journal on Computing*, 2(1):4–32, 1989.
- [20] H. Hoppe. Progressive meshes. In *Siggraph '96*, pages 99–108. ACM Siggraph, 1996.
- [21] R.E. Kass and A.E. Raftery. Bayes factor. *Journal of the American Statistical Association*, 90:773–795, 1995.
- [22] W.E. Lorensen and H.E. Cline. Marching cubes: a high resolution 3d surface algorithm. In *Computer Graphics*, volume 21, pages 163–169. ACM Press, july 1987.
- [23] B.S. Morse, S.N. Pizer, and C.A. Burbeck. General shape and specific detail: context-dependent use of scale in determining visual form. In *Second International Workshop on Visual Form*, pages 374–383. World Scientific, 1994.
- [24] S. Muraki. Volumetric shape description of range data using “blobby model”. In *Computer Graphics*, volume 25, pages 227–235. ACM Press, July 1991.
- [25] M. Näf, O. Kübler, R. Kikinis, M.E. Schenton, and G. Székely. Characterization and recognition of 3d organ shape in medical image analysis using skeletonization. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, San Francisco (USA), June 1996.
- [26] I. Pitas and A.N. Anastasios. Morphological shape decomposition. *IEEE PAMI*, 12(1):38–45, 1991.
- [27] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C, the Art of Scientific Computing*. Cambridge University Press, 1992.
- [28] A. Ricci. A constructive geometry for computer graphics. *The Computer Journal*, 16:157–160, May 1973.
- [29] J. Rossignac and P. Borrel. Multi-resolution 3d approximation for rendering complex scenes. In B. Falcidieno and T.L. Kunii, editors, *Modelling in Computer Graphics*, pages 455–465. Springer Verlag, 1993.
- [30] W. Schroeder, J. Zarge, and W. Lorensen. Decimation of triangle meshes. In *Siggraph '92*, volume 26, pages 65–70. ACM Siggraph, July 1992.
- [31] S. Sclaroff and A. Pentland. Generalized implicit functions for computer graphics. In *Computer Graphics*, volume 25, pages 247–250. ACM Press, July 1991.
- [32] S. Sclaroff and A. Pentland. Model matching for correspondence and recognition. *IEEE PAMI*, June 1995.
- [33] A.J. Scott and M.J. Symons. Clustering methods based on likelihood ratio criteria. *Biometrics*, 27:387–397, 1971.
- [34] E.P. Simoncelli. A rotation invariant pattern signature. In *IEEE International Conference on Image Processing*, volume 3, pages 185–188, Lausanne (Switzerland), September 1996.
- [35] B. Soroka, R. Andersson, and R. Bajcsy. Generalized cylinders from local aggregation of sections. In *Pattern Recognition*, volume 13, 1981.
- [36] G. Taubin. Distance Approximation for Rasterizing Implicit Curves. *ACM Transactions on Graphics*, 13:3–42, 1994.
- [37] G. Taubin, F. Cukierman, S. Sullivan, J. Ponce, and D.J. Kriegman. Parameterized families of polynomials for bounded algebraic curve and surface fitting. *IEEE PAMI*, 6:287–303, March 1994.
- [38] C. Lefèvre V. Burdin, C. Roux and E. Stindel. Modeling and analysis of 3-d elongated shapes with application to long bone morphometry. In *IEEE Transactions on Medical Imaging*, volume 15, pages 79–91. IEEE, February 1996.
- [39] O. G. Okunev V. V. Savchenko, A.A. Pasko and T. L. Kunii. Function representation of solids reconstructed from scattered surface points and contours. In *Eurographics '95*. Blackwell Publishers, 1995.

Synchronisation Formalism, Resource and Plant Models for Plant Ecosystem Simulation

Vincent Le Chevalier
Ecole Centrale Paris - MAS
INRIA Saclay, EPI DigiPlante
Chatenay Malabry, F-92295, France
vincent.le-chevalier@ecp.fr

Marc Jaeger
CIRAD - AMAP
INRIA Saclay, EPI DigiPlante
Montpellier, F-34398, France
marcjaeg@gmail.com

Paul-Henry Cournède
Ecole Centrale Paris - MAS
INRIA Saclay, EPI DigiPlante
Chatenay Malabry, F-92295, France
paul-henry.cournede@ecp.fr

Abstract

Vegetation ecosystem simulation, at the landscape or crop level, is a challenging topic involving multidisciplinary aspects. Several models of biophysical components must be coupled properly in order to obtain a meaningful simulation. This also raises interesting problems in software architecture.

In this paper, we first outline a synchronisation formalism and software architecture adapted to the simulation of systems built from several coupled models. The models are linked by caches that provide inputs and receive outputs, and control the time step of the simulation. Next, we describe specific models for resource management and plant growth, that are essential for landscape simulation. The resource models are designed to solve transparently the problem of competition. The implemented plant model involves three environmental parameters (temperature, light, water) and uptakes water according to plant growth. Finally, these models are coupled in simulations of a plant interacting with water resource, showing various complex behaviours in time.

1. Introduction

Our goal is to implement what we call a *functional landscape*: a simulation and visualisation of a biophysical landscape, showing complex spatial and temporal dynamics. Such simulations are typical complex system simulations, exciting the interest of research communities with a large panel of tools and approaches [1], [2].

More precisely, we focus only on plant ecosystems and their interaction with the environment. This has been explored previously in two papers. In [3] a simple prototype was presented, showing the technical feasibility of the approach. In [4], more realistic models were implemented in a better structured code. Nevertheless these first prototypes highlighted some unforeseen challenges in simulation.

We describe biophysical landscapes as a set of coupled models that would ideally be designed by independent

experts in their fields (plants, hydrology, climate science, etc.). Once coupled together, these models should exhibit emerging properties at the landscape scale, in space and time. Thus the landscape is built by linking together several models that are relatively independent from one another, without any of them having the complete knowledge of the whole landscape. Different models of the same landscape component could also be swapped in order to be tested and compared.

This implies a very modular approach (called *model composability* [5]), which was not completely respected in the first prototypes. Moreover, the synchronisation of the landscape processes in time was inconsistent and laid constraints on what was possible to model. For example, competition between several plant species was impossible to implement in a clean realistic fashion. Software architecture had to be radically rethought to solve these problems.

The DEVS (Discrete EVents Specification) formalism [6], [7] is a generic approach to the building of systems consisting of communicating models. It has been used in the context of landscapes, in conjunction with other description formalisms as reviewed in [8]. However, it is not necessarily the most intuitive for modelers to use or the most adapted to the description of biophysical processes, and its implementations (see [9]) are not always practical to use or extend depending on the software configuration and development resources at hand. This has led many authors to develop other simulation formalism and architecture (e.g. [2]).

The main drawback of the DEVS formalism is that it is based on discrete events, which are not really the natural description of landscape processes in our approach. We present in the first part of this paper an alternative formalism that is more focused on the simulation data, with models acting on the data and changes in the data triggering the computation of models. This separation of models and data allows for an easier synchronisation and facilitates simulation dumps. It is also well suited to the development of models operating on spatially heterogeneous data.

In the next part specific models necessary for landscape simulation are described. We start by presenting models

for resource management (particularly useful for water), that attempt to solve elegantly the problem of competition. Then a plant model is introduced, modified to cope with the particular constraints of the simulation architecture. It interacts with the environment through input parameters such as temperature, light and soil water, and uptakes water proportionally to the biomass produced by photosynthesis. Finally, a sample simulation is explored including all these models, showing promising capacities for eventual full-landscape simulations.

2. Synchronisation formalism

The software architecture uses mainly two kinds of objects: *Models* and *Caches*. The basic idea is that Models do all the computation, and Caches realize the abstraction of data. Models read their inputs in Caches, write their outputs to other Caches. Caches then update the data in a synchronous fashion. This basic architecture is represented on figure 1. The essential property of this architecture is that the simulation data is stored in a separate data structure, which can always be kept in a consistent state (assuming the update is instantaneous). Not only does this allow for a more consistent simulation, it also solves the problem of simulation dumps that permit to resume a previous interrupted run.

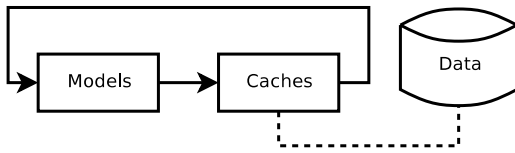


Figure 1. The basic architecture. The arrows represent the flow of information (Models compute new values for Caches, Caches are updated giving new inputs to Models). The dotted line represent the link from Caches to the actual data, that is kept separate from Models.

2.1. Caches

Caches are responsible for the actual evolution of the data. They have a link to an exterior data structure containing the variables of the simulation, and temporarily hold the result of the computations of the Models, hence their name.

Caches communicate with Models via two specific interfaces, exhibiting different behaviours, to enforce the synchronism of the system's evolution.

The first one is the *Getter*. A Model that has a link to a Getter is able to retrieve a value from it. The Cache, in this case, returns the value currently stored in the external data structure. The second one is the *Setter*. Through a link with a Setter, a model can specify the future value of the data. The Cache copies it and keeps it until the next update.

Because these two behaviours are kept separate, there is no risk that the value is modified during the computation of the Models. Normally, the same Getter can be linked to several Models, but one Setter should only ever be linked to one Model.

The value associated to the Cache can be updated in several ways depending on the problem at hand. The most obvious way is to simply copy the new value in the exterior data structure. However, in certain cases, holding two copies of the entire data type is wasteful memory-wise. Another way to proceed in these situations is to create the Cache as a Getter of the 'big' data type, and a Setter of a 'variation' data type, that contains just the changes needed to update the data. The benefits of this update strategy can be important when the changes computed can be compressed or affect only a small portion of the complete data structure. It can also be a natural choice when a model computes a variation of its output, and not a value.

A Cache is said to be in a *consistent* state when the update would not cause any significant change in the value. This definition is important to judge the convergence of the system in certain cases. Whether the change is significant or not is determined based on the update method, the type of the value, and a global threshold.

2.2. Models

Models are responsible for all the computations, and for the synchronisation of their optional internal states. Some models can be *stateless*, when the result of their associated computations depends only on the values of the inputs.

Normally, no raw data should be found in a Model. Any data should be accessed through Caches. This is necessary because nothing guarantees that the model itself will persist in memory during the whole simulation. Only the associated state data, linked to the Model's Caches is persistent.

Models define two computation functions, each affecting two subsets of their outputs. First there is a cumulative function:

$$O_C(t) = \text{cumul}(t, t_{prev}, I(t_{prev})) \quad (1)$$

where O_C is the list of cumulative outputs, t is the current time, t_{prev} is the time of the previous call to *cumul*, and I is the list of the inputs.

This function is responsible for keeping the memory of what happened during the time interval $[t_{prev}, t]$, during which the inputs I are assumed to be constant. Indeed it is a form of the basic Euler integration method. Secondly, there is a transitory function:

$$O_T(t) = \text{trans}(I(t)) \quad (2)$$

where O_T is the list of transitory outputs.

This function allows to specify models whose outputs change instantly when the inputs change. A typical example

would be an additionner that adds together two inputs. Such a model does not have any memory and just reacts to changes in inputs.

Aside from computations driven by changes of their inputs, models can specify a time at which a call of their cumulative function must happen :

$$t_n(t_{prev}) = time(t_{prev}, I(t_{prev})) \quad (3)$$

The dates at which the system state is computed are based on the results of this function for each model. In other words, the models themselves are responsible for the global time step of the simulation.

How these functions can be used to solve the evolution of the system is explained in the next section.

2.3. Hierarchies and simulation Management

Models and Caches can be nested in a hierarchy. This is particularly useful in order to model multiple time scales and to implement more efficient spatially distributed algorithms.

It would be unsafe to allow a direct link from a sub-Model to the outside of the Model as this would potentially lead to inconsistency in the synchronization; instead special non-persistent Getters and Setters are defined, that serve as buffers between the sub-Models and the inputs. The external data is nested in a way that reflects the Model hierarchy. All this architecture is schematically represented in figure 2.

A simulation Manager is a particular type of Model that has no input and no output, and just synchronizes the computations of all its internal sub-Models. It holds a list M of sub-Models m_j , a list C of Caches, and a representation of the graph formed by connecting the Models to the Caches.

The *trans* function of the Manager is empty, and its *time* function returns infinity. The *cumul* function advances the simulation from t_0 to t_1 in several steps, depending on the results of the calls to the *time_j* functions of the sub-Models.

```

⟨cumul( $t_1, t_0$ )⟩ ≡
   $t_c \leftarrow t_0$ 
   $t_p \leftarrow t_0$ 
  do
    ·  $t_c \leftarrow \min(\{time_j(t_p, I_j)\}_j, t_1)$ 
    · ⟨Time advance⟩
    · ⟨Convergence⟩
    ·  $t_p \leftarrow t_c$ 
  while( $t_c < t_1$ )

```

Here t_c is the current time of the simulation, and t_p is the previous time, used when the function *cumul_j* is called for each model in the *Time advance* part. We call *cumul* for each sub-Model, and then update the Caches:

```

⟨Time advance⟩ ≡
  for all models in  $M$ 
  ·  $O_j^C \leftarrow cumul_j(t_c, t_p, I_j)$ 
  end
  ⟨Update the Caches⟩

```

In the updating of the Caches, a Cache in a consistent state is skipped. A boolean *done* is used to determine whether a Cache has been updated or not, and will be checked in the next stage detailed below (*Convergence*).

```

⟨Update the Caches⟩ ≡
  done ← true
  for all caches in  $C$ 
  · if the cache is not consistent
  · · update the cache
  · · done ← false
  · end
  end

```

The last stage is the convergence of the system of sub-Models and Caches to a consistent state, thanks to the *trans* functions of the Models. Basically the *trans* functions of the models are repeatedly called, until all the Caches are consistent.

```

⟨Convergence⟩ ≡
  while done is not true
  · for all models in  $A$ 
  · ·  $O_j^T \leftarrow trans_j(I_j)$ 
  · end
  · ⟨Update the Caches⟩
  end

```

The whole *cumul*(t_1, t_0) function is generic and does not depend on the contents of the models. It is fairly easy to generalize the principle to a Manager that would have inputs and outputs as well, in order to build a hierarchy of Managers.

Of course the pseudocode described here is fairly sub-optimal and repeats exactly the same computation a number of time. The actual implementation keeps track of which Caches change and which Models should be computed, however it gives the same result and follows the same principles.

3. Application to biophysical models

3.1. Resource management

Within bio-physical landscapes, one of the essential phenomenon we wish to simulate is the competition for resources. Water is the most obvious example. To this end, we had to specify the interaction of various Models with resources, either supplying them or acting on them.

Interaction with resources takes the form of three complementary inputs and outputs, depending on whether the model allows actions on the resources or performs actions on the resources. A model that allows actions on a resource has two outputs, the quantity of resource *available* for uptaking and the quantity of resource *free* for filling, and one input, the rate at which the resource is taken, called *demand* henceforth. This pattern of input and outputs is called a Provider in the rest of this work. Conversely, a model that

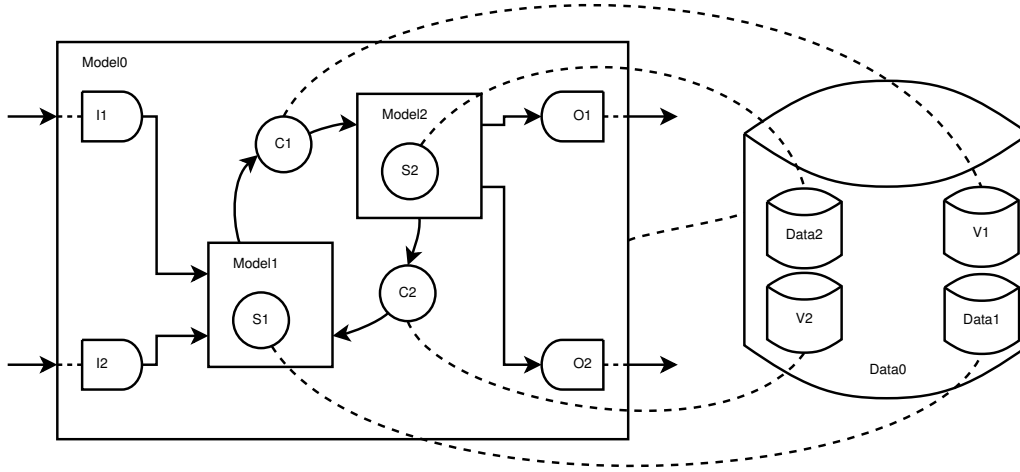


Figure 2. Hierarchical nesting of models and caches. Model0 has a State structure represented by Data0. Model0 contains sub-Models, Model1 and Model2, linked through Caches, C1 and C2. Each of these has associated data included within Data0. Input buffers, I1 and I2, and output buffers, O1 and O2, allow communication with the outside.

performs actions on a resource has two inputs, the quantities available and free, and one output, the demand. This pattern of inputs and output is called an Accessor in the rest of this work.

Both patterns of inputs or outputs can be noted as triples (D, F, A) where D is the demand, F is the free space that can still be filled with resource, A is the available resource that can be taken.

Containers are the basic Providers that hold the resources at the landscape level. They are meant to be independent of the processes acting on the resources. Their limitations are the chief cause of competition. Several processes can compete over the same container thanks to other models called Dispatchers, further detailed below.

The state of a Container is a structure comprising the current quantity of resource Q , as well as the minimum and maximum quantity that can be held in the container (Q_{min} and Q_{max} respectively).

Containers are Providers and have the input previously cited, the demand D . They have two capacities as outputs, the available quantity of resource A , and the free quantity F . Both of these outputs are transitory.

The cumulative function just updates the state according to how much demand there has been on the time interval:

$$Q = Q_{prev} - D(t - t_{prev}) \quad (4)$$

The transitory function computes A and F according to the state:

$$A = Q - Q_{min} \quad (5)$$

$$F = Q_{max} - Q \quad (6)$$

The time function returns the time before the limits Q_{min} or Q_{max} are reached with the current demand:

$$t_n = \begin{cases} t_{prev} + \frac{Q - Q_{min}}{D} & \text{if } D > 0 \\ t_{prev} + \frac{Q - Q_{max}}{D} & \text{if } D < 0 \\ \infty & \text{if } D = 0 \end{cases} \quad (7)$$

The Dispatchers are communication components that allow several objects to interact with several Containers. They contain Providers that act as kind of virtual Containers, and Accessors that are linked to actual Containers.

A Dispatcher has no state. Given that a Dispatcher is a pure transitory model, it does not have evolutions of its own. Thus $t_n(t_{prev}) = \infty$, and the cumulative function does nothing. All the outputs are transitory.

Let us consider a Dispatcher with N Providers $\{(D_i^p, F_i^p, A_i^p)\}_{i \in \{1, \dots, N\}}$ and M Accessors $\{(D_j^a, F_j^a, A_j^a)\}_{j \in \{1, \dots, M\}}$. A list of coefficients (γ_{ij}) is used to represent the strength of the relation between Provider i and Accessor j .

All told, Dispatchers have $MN + N + 2M$ inputs: γ_{ij} , D_i^p , F_j^a , A_j^a , and $2N + M$ outputs: F_i^p , A_i^p , D_j^a .

The transitory function uses weighted sums to transfer information back and forth between the Providers and the Accessors. This is only one of the many possible ways to build this communication, but it is particularly well suited to water resource. The computation involving the demand is the most complex, as cases must be distinguished in order to balance the load on the different Accessors. The formulas

used are:

$$F_i^p = \sum_j \gamma_{ij} F_j^a \quad (8)$$

$$A_i^p = \sum_j \gamma_{ij} A_j^a \quad (9)$$

$$D_j^a = \sum_{i, D_i^p > 0} D_i^p \frac{\gamma_{ij} A_j^a}{A_i^p} + \sum_{i, D_i^p \leq 0} D_i^p \frac{\gamma_{ij} F_j^a}{F_i^p} \quad (10)$$

They ensure the conservation of demand and act more on the Accessors that have the greatest availability of resource.

3.2. A continuous GreenLab Model

The GreenLab model has been used as a basis for modeling plant in the framework presented. This model has shown promising capabilities, for example it can be calibrated to reproduce the behaviour of real plants [10], while keeping a relatively low complexity thanks to structural factorization [11]. This makes it especially interesting if we want to realistically run simulations at the landscape level. However, we had to modify its expression since its definition is based on growth development cycles and not on calendar time.

3.2.1. General principles of a continuous GreenLab plant model. The plant P is represented, as far as functioning is concerned, as a list of metamers m_i , each holding the masses of the organs they contain by compartment $[M_i^1, \dots, M_i^c]$, c being the number of compartments considered for the plants. More auxiliary data is associated to a metamer, but the accurate description is outside the scope of this article.

Plant development (the apparition of new organs) is a discrete process driven by changes in thermal time, as is the case in the classical GreenLab model. The development cycle σ of a plant is defined as:

$$\sigma = \int_0^t \frac{\max(0, T - T_b)}{T_g} d\theta \quad (11)$$

where t is the current time, T_b is the base temperature below which development pauses and T_g is the number of degree-day necessary to complete a growth cycle. The organogenesis process adds metamers to the list when σ takes an integer value, computing their mass at apparition if necessary.

Between the organogenesis events, the plant continuously produces biomass by photosynthesis and allocates it to the compartments of the metamers. This is the key difference with the classical GreenLab model, where production is computed at the level of the whole cycle and allocated during the next cycle.

The biomass flux generated by photosynthesis is expressed as:

$$Q(t) = E(t) \alpha S_p \left(1 - \exp \left(-k \frac{S(t)}{S_p} \right) \right) \quad (12)$$

where $S(t)$ is the leaves surface computed from their masses, $E(t)$ is a parameter representing the environment influence, k is the coefficient of the Beer law, and α , S_p are plant parameters.

This rate of biomass production is assumed to be available for the growth of all organs in the plant. It is allocated proportionally to the sink strength of each organ, given by $s_i^o(\sigma)$ for the compartment o of metamer i . First the total sink is computed as $D(t) = \sum_i \sum_{o=1}^c s_i^o(\sigma(t))$. Then the time derivative of metamer i can be computed as:

$$\frac{dm_i}{dt} = \left[\frac{dM_i^1}{dt}, \dots, \frac{dM_i^c}{dt} \right] = \frac{Q(t)}{D(t)} [s_i^1(\sigma), \dots, s_i^c(\sigma)] \quad (13)$$

Along with the derivative of the cycle, which is trivial to express from equation (11), this last equation (13) allows us to formally find the derivative of the plant $\frac{dP}{dt}$ at time t as a function of the plant P , plant parameters and environment. Several methods can be used to simulate such a system, a simple Euler integration is used in the architecture described above.

3.2.2. Inserting the continuous GreenLab into a proper Model. The plant Model contains two state variables, accessed through owned Caches as described above: the state of the plant P , and its derivative $\frac{dP}{dt}$. Remember that the development cycle σ and its variation is included in these. It has Getters for the parameters and for three environment variables: the incident radiation R , the temperature T and the soil hydric potential ψ . Finally, the only transitory Setter is the rate of water uptake in the soil W .

The main computation done in the transitory function of the Model is the environment value E appearing in (12). It is modeled as the product of three factors:

$$E = R E_T E_w \quad (14)$$

where E_T is the effect of temperature and E_w is the effect of soil water, that has already been described in our article [4].

A simple function was chosen for E_T , that has a maximum of 1 at an optimal temperature T_{opt} and slopes to zero at two extreme temperatures T_b (the base temperature already mentioned) and T_h :

$$E_T = \left(\left(\frac{T - T_b}{T_{opt} - T_b} \right)^{\frac{T_{opt} - T_b}{T_h - T_b}} \left(\frac{T_h - T}{T_h - T_{opt}} \right)^{\frac{T_h - T_{opt}}{T_h - T_b}} \right)^\gamma \quad (15)$$

The parameter γ controls the accuteness of the maximum. A more biologically sound model should be defined, but this is sufficient to test our simulation framework. From the E value, the transitory function performs the computation of $\frac{dP}{dt}$ and W . That last value is taken as proportional to the rate of biomass production:

$$W(t) = \frac{Q(t)}{wue} \quad (16)$$

where wue is the water use efficiency of the plant.

Organogenesis is also done in the transitory function.

The cumulative function only updates the plant state with a simple Euler integration step:

$$P = P_{prev} + (t - t_{prev}) \frac{dP}{dt} \quad (17)$$

Lastly, the time limit function computes the time of the next organogenesis step, assuming a constant temperature:

$$t_n = (\sigma_{next} - \sigma) \frac{T_g}{\max(0, T - T_b)} \quad (18)$$

where σ_{next} is the development cycle of the next organogenesis event. It uses the expression given in (11).

3.3. The plant Model in interaction with its environment

In this section we describe a small test-case that allows to showcase the potential of the approach described so far. A simple simulated system is built that models the interaction between the plant model and the water resource.

The system comprises one Plant model and one Container, as this is the bare minimum needed to model plant and soil water. To prevent water shortage, an irrigation Pump is linked to the soil Container. Various behaviours of this pump could be modeled, here the Pump provides water at a fixed rate K_p , stops when the stock is above a given percentage (90% here), and starts again if it goes below another percentage (10% here). This is a rough model of a common irrigation strategy, based on soil water content rather than on plant state. The strategy is voluntarily made extreme and simplistic in this example; a real-life irrigation strategy probably should not let soil water content fall down so much.

We choose a very smooth plant model, without branches, that just emits one metamer (one leaf and one internode) at each development cycle. The Plant and the Pump act on the same Container, through a Sharer which is a specialized Dispatcher. The soil hydric potential is computed from the stock in the soil Container using the following formula:

$$\psi = \frac{1}{\lambda} \log\left(\frac{A}{A + F}\right) \quad (19)$$

where λ is a soil parameter, and A, F are the available resource and free space in soil, respectively.

A diagram of the whole system is represented on figure 3. Note that in order to compute its t_n function the pump also has to read the total uptake of soil water.

We know from previous studies of the GreenLab model in isolation that a steady state is reached in a constant environment (see chapter 4 of [12]). However coupling it with the pump and container revealed other more complex behaviours. To illustrate these we present on figure 5 several plots comprising three curves: the irrigation rate (0 means

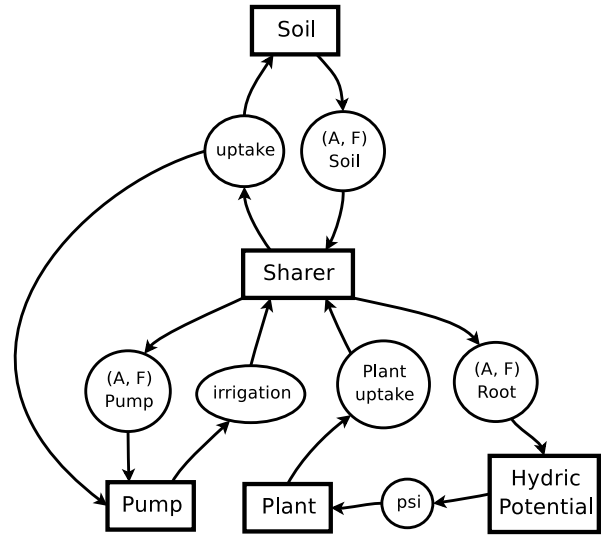


Figure 3. The system used in the test-case. Models are represented by rectangular nodes, important Caches are represented as ovals. Note the central role of the Sharer, that makes it possible for the Pump and the Plant to interact with the same Soil. The same principle could be adapted to include more plants and pumps, as well as more containers for the Soil.

that the pump is off), the plant water uptake (proportional to the biomass production), and the plant's leaf surface.

After roughly similar transitory phases, two kind of asymptotic behaviours can be observed.

- when K_p is low, the plant uptake and the pump irrigation compensate in the steady state. The pump is constantly on. The leaf surface is mostly stable, it oscillates slightly because of the leaves' death. This case is represented at the top of figure 5.
- when K_p is high, the plant uptake cannot compensate the pump irrigation. Therefore the saturation point of the pump is reached, the pump stops, the water content diminishes causing a diminution in production. Once the low water content is reached the pump starts again, and the production augments. The leaf surface also oscillates with a small phase shift, because the effect of the production are integrated in the leaf surface. This case is represented on the second plot of figure 5.

Generally, in the first case the final leaf surface augments when K_p augments, and in the second case the final leaf surface augments when K_p diminishes. There is a limit case that marks the boundary between low K_p and high K_p , where the system abruptly switches from one asymptotic behaviour to another. We illustrate here a perfect example of the consequences of the coupling of strongly non-linear models; a bifurcation appears, that changes dramatically the result for very small variations of the parameter K_p . This

behaviour is illustrated on figure 4. The bottom plot of figure 5 shows what is found around the limit value of K_p : the plant seems to be in the oscillatory state, until plant water uptake exactly compensates irrigation. Then the production and leaf surface rise suddenly and remain stable.

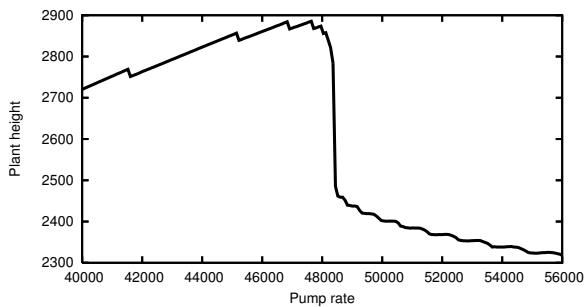


Figure 4. Plant height versus pump rate K_p . The abrupt change of behaviour is readily visible. The small discontinuities when K_p is low are due to a variation in the number of irrigation impulses that the plant has seen during the early growth stage.

4. Conclusion

A new simulation formalism and architecture has been presented. It solves appropriately the synchronization problems that are encountered when trying to simulate a multi-model system. Model and data are kept separate and the formalism facilitates the expression of biophysical models. It allows the modeler to focus on the logical description of the system and on the behaviour of each model, without the need to care about the ordering of computations or the appropriateness of the time step for all the models. Of course it is still possible that a model blocks the simulation, which means that its behaviour is not appropriate and that it needs to be refactored. The example application shows that complex behaviour can quickly emerge even in apparently simple systems.

An important lesson is that even though the simulation architecture has been made as flexible and generic as possible, some models still need to be modified because they were never designed to interact realistically with the more global system. The plant model is an example of this kind of situation. We expect to find more cases as we implement full-scale landscapes.

Further work will include a more complete landscape simulation, or at the very least more applications with realistic bio-physical models at the crop and small landscape scales. Preliminary tests have shown that the simulation architecture can cope with spatialised models with relative ease. These bigger spatialised simulations will be an opportunity for us to explore the potential of the architecture

for parallel computing, but several models will have to be re-implemented first.

The simulation architecture could be perfected when accuracy becomes a concern; specifically, other temporal integration schemes should be implemented in order to bring more precision and stability to the simulation.

Lastly, from the point of view of software engineering, the architecture also has some potential that would have to be exploited. Several tedious steps could be automated using code generators, preventing some basic errors that are sometimes difficult to detect.

5. Acknowledgements

This work is supported by the TERRADATA project (Pôle de Compétitivité Cap Digital), the ANR CIS07 3Worlds project, the Agropolis Fondation RTRA 0803-007 and RTRA 0803-021 projects.

References

- [1] D. C. Parker, S. M. Manson, M. A. Janssen, M. J. Hoffmann, and P. Deadman, "Multi-Agent Systems for the Simulation of Land-Use and Land-Cover Change: A Review," in *Annals of the Association of American Geographers*, 2002.
- [2] R. Costanza and A. Voinov, *Landscape Simulation Modeling: a Spatially Explicit, Dynamic Approach*, ser. Modeling Dynamic Systems. Springer, 2004.
- [3] V. Le Chevalier, M. Jaeger, X. Mei, A. Lesluye, and P.-H. Cournède, "A functional landscape prototype to simulate water resource competition between plants," in *PMA06 - Plant growth Modeling, simulation, visualization and their Applications. November 13-17, 2006, Beijing, China*, T. Fourcaud and X. Zhang, Eds. IEEE Computer Society, Los Alamitos, California, 2007, pp. 124–131.
- [4] V. Le Chevalier, M. Jaeger, X. Mei, and P.-H. Cournède, "Simulation and visualisation of functional landscapes: Effects of the water resource competition between plants," *Journal of Computer Sciences and Technology*, vol. 22, no. 6, pp. 835–845, november 2007.
- [5] P. C. Davis, P. A. Fishwick, M. C. Overstreet, and D. C. Pegden, "Model composability as a research investment: responses to the featured paper," in *WSC '00: Proceedings of the 32nd conference on Winter simulation*. San Diego, CA, USA: Society for Computer Simulation International, 2000, pp. 1585–1591. [Online]. Available: <http://portal.acm.org/citation.cfm?id=510610>
- [6] B. P. Zeigler and S. Vahie, "Devs formalism and methodology: unity of conception/diversity of application," in *In Proceedings of the 25th Winter Simulation Conference*. ACM Press, 1993, pp. 573–579.
- [7] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of Modeling and Simulation*. Academic Press, January 2000.

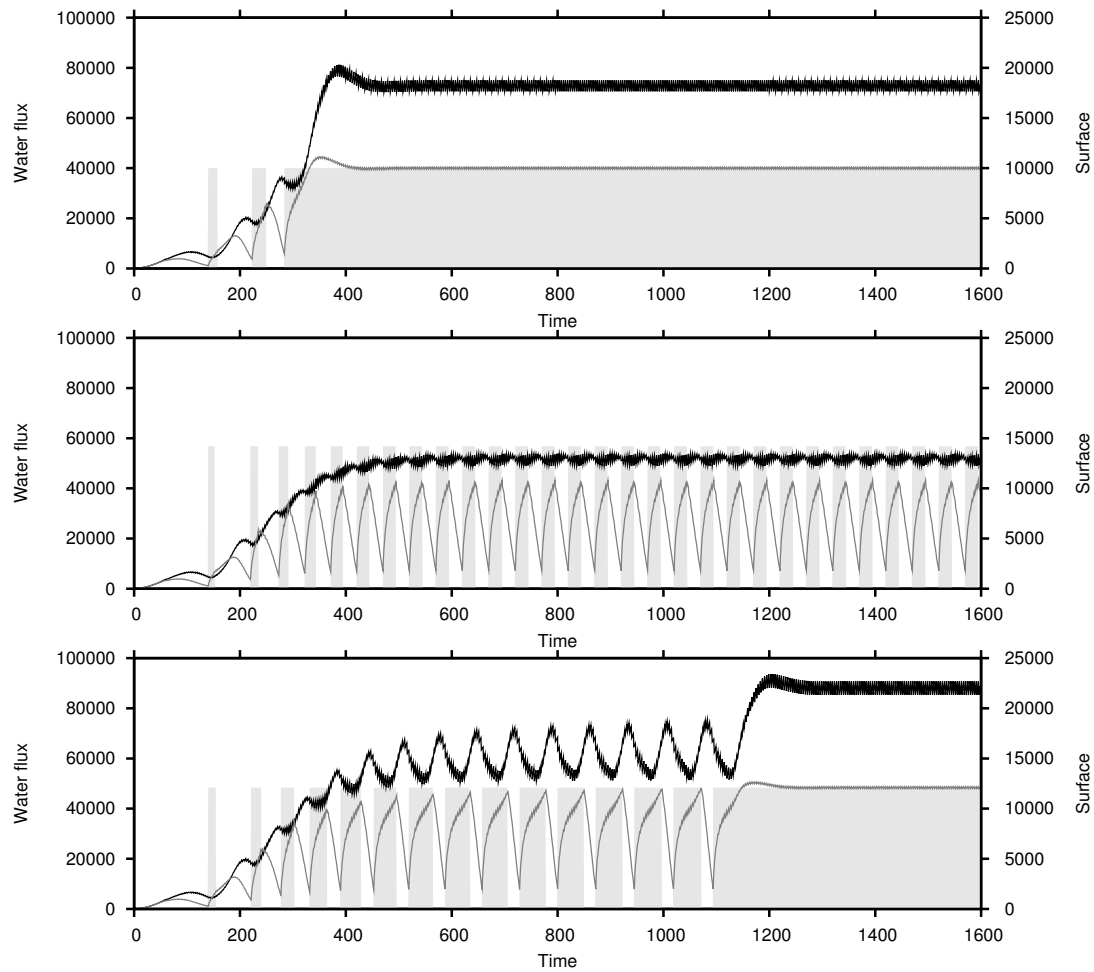


Figure 5. Several plots showing the evolution of the system along time for different values of the pump rate K_p . The black line is the leaf surface, the gray line is the plant water uptake, the light gray bars represent impulses of the irrigation pump. On the top graph K_p is low, in the steady state the pump stays on permanently and the irrigation is exactly compensated by the plant water uptake. On the middle graph K_p is high, the plant cannot compensate for the irrigation and an oscillatory state is reached. There is a clear bifurcation between these behaviours, within a very small range of K_p . The bottom graph corresponds to a limit value for K_p and shows how the system switches from one behaviour to another before reaching its steady state.

- [8] G. R. Mayer and H. S. Sarjoughian, "Complexities of simulating a hybrid agent-landscape model using multi-formalism composability," in *SpringSim '07: Proceedings of the 2007 spring simulation multiconference*. San Diego, CA, USA: Society for Computer Simulation International, 2007, pp. 161–168.
- [9] G. Quesnel, R. Duboz, and E. Ramat, "The Virtual Laboratory Environment – An operational framework for multimodelling, simulation and analysis of complex dynamical systems," *Simulation Modelling Practice and Theory*, vol. 17, pp. 641–653, April 2009.
- [10] Y. Guo, Y. Ma, Z. Zhan, B. Li, M. Dingkuhn, D. Luquet, and P. de Reffye, "Parameter optimization and field validation of the functional-structural model greenlab for maize," *Annals of Botany*, vol. 92, no. 2, pp. 217–230, 2006.
- [11] P.-H. Cournède, M.-Z. Kang, A. Mathieu, J.-F. Barczi, H.-P. Yan, B.-G. Hu, and P. de Reffye, "Structural factorization of plants to compute their functional and architectural growth," *Simulations*, vol. 82, no. 7, pp. 427–438, 2006.
- [12] V. Letort, "Adaptation of the greenlab growth model to plants with complex architectures and multi-scale analysis of source-sinks relationships for parameter identification," Ph.D. dissertation, Ecole Centrale Paris, 2008.

EFFICIENT VIRTUAL PLANT DATA STRUCTURE FOR VISUALIZATION AND ANIMATION

Marc Jaeger

*CIRAD UMR AMAP, EPI Digiplante (joint INRIA, CIRAD, Ecole Centrale Paris project team)
TA A51 / PS2, 34398 Montpellier Cedex 5, France*

Ruoxi Sun and JinYuan Jia

*College of Software Engineering, Tongji University
Shanghai, China*

Vincent Le Chevalier

*Ecole Centrale de Paris, MAS Laboratory, EPI Digiplante (joint INRIA, CIRAD, Ecole Centrale Paris project team)
Grande Voie des Vignes, 92295 Châtenay-Malabry Cedex France*

ABSTRACT

Virtual plants are interesting but costly objects in video games and virtual reality applications. Despite advances on Level of Detail approaches and increasing GPU power, previous work fails in producing convincing and generic vegetation with low bandwidth requirements for web display. The efficiency drops down considering costs of compression and inner structure redundancy is still poorly explored.

We propose here a tree graph as a generic output of simulation, easy to implement from any rule based plant generator. This graph allows to efficiently compress similar structures, that appear naturally as a result of procedural definition. Nodes carry minimal attributes, mainly related to age. Upon display requests, nodes orientation and precise geometry are computed using a simple graph exploring approach based on a very small number of parameters. This exploration allows to generate all past growth stages and dynamic effects such as pruning. We show how this graph can be efficiently generated from L_systems, as well as from a functional-structural plant simulator. We illustrate that, thanks to this encoding and reconstruction scheme, a simple tree simulation can generate a wide range of geometrical plant exports at various ages, and is compatible with classical level of detail rendering approaches for geometric models.

KEYWORDS

Computer Graphics, Virtual plants, natural scene rendering, L systems, FSPM models

1. INTRODUCTION

In video games and virtual reality applications, natural object visualization and transfer of geometrical data are still a challenging topic, despite all the technological progress on graphics cards performance. This is especially the case of scenes composed by virtual plants seen from close views. In the past decade, a wide range of studies were held on the subject, mainly based on significant decrease of plant geometry, replacing complex geometrical structures by simple ones according to viewer distance. The main distinction found in literature is between image based approaches and compressed polygonal approaches. Among image based approaches, one can note the use of impostors [Max 1999]. Impressive leaf cluster impostors examples and an interesting review of various techniques used for realistic single tree visualization can be found in [Fernandez 2007]. Images approaches show potential for fast rendering but suffer from relative complexity on close view due to the multiplication of textures, thereby incurring high bandwidth costs if Web visualization is the target. Many authors did also work on polygonal representations, simplifying the tree structure and its mesh conversion, with specific methods on foliage [Deng 2007]. An overview of such methods is given by [Deng 2010]. However, sets of level of details models -multi resolution models or just billboards- introduce costly pre-processing constructions. And, once again, for Web visualization, sets of multi-scale representations lead to high bandwidth costs if realistic close and far views are requested on the

client display. Finally, those techniques seldom address growth animation. In a natural scene with plants of the same species with various ages, such techniques, working at a low semantic level, show poor reusability. Several authors show that efficient tree generation can drop down generation complexity from exponential to linear, with automates [Yan 2002] or L-systems [Brownbill 1998], but those approaches were not explored for data transfer and rendering. While compression using grammars is popular in data analysis [Nevill-Manning 1994], it is seldom used for the encoding and transfer of virtual object.

In order to visualize natural scenes on the Web, an efficient solution consists in sending to the client the construction rules and the plant generator. Such approaches were already adopted for plants reconstructed from photographs [Sun 2009] with L-systems. But it is easy to imagine that such simulators may not be simple, low cost and compact software pieces. L-systems [Prusinkiewicz 1990] are widely used in literature, however, as described by [Deussen 2004], various techniques are currently available and popular (automates, patterns [Deussen 1997], specific grammars, fractals, ...), leading to lots of specific tools on the client.

The idea we want to push here is that, for a wide range of tree plant generators, there may be intermediate structures of interest between rules (specific to each approach) and the final geometrical structure rendered at full scale (usually a oriented tree graph). We present here a tree graph, a structure open to most plant generators. We show that, if we take care of basic botanical constraints, this structure can drastically compress duplicated patterns in the tree, and can be used to generate all past growth steps of the plant. As applications, we also propose a reconstruction frame allowing to generate the growth steps geometry and show its potential on natural scenes, using tree graphs generated by different simulators.

In the next section, we detail how we encode tree structures taking advantage of redundancies deduced from biological typologies. In the third section, we show how this coding can reconstruct the geometry of plants at several growth stages. The last section illustrates example applications with their performances.

2. EFFICIENT TREE ENCODING, THE TREE GRAPH

We remind here some basic elements of rule based vegetation. Our approach uses the plant modeling formalism GreenLab [De Reffye 2003], implemented with a L_system [Prusinkiewicz 1990]. We describe then the tree graph with its topology and specify a set of node attributes.

2.1 Efficient Tree Structure Generation

As described by O. Deussen [Deussen 2004], rule based plant generation applies rules on an initiator, whatever the underlying technique is. However, to generate a wide variability of shapes, simulators must allow for variability in both the structure generation (the tree topology) and the geometry (angles), leading to a set of rules. Another important point is the age, or the number of cycles during which the rules are applied in the course of a simulated growth process. Based on these concepts, a L_system production is implemented that uses the notion of substructures, as defined by Yan et al [Yan 2002], minimizing redundancies.

2.1.1 Basis of Procedural Tree Generation: The Metamer as a Basic Node

In botany, plant architecture is defined from a typology of various axis characters [Barthelemy 2007]. Tree architecture is build from axis, defined by growth units (annual shots in temperate trees), themselves defined by nodes. Each node carries a leaf from which a lateral bud can grow a new axis. The node with its leaf is called a metamer and thus defines the basic topological unit.

The plant structure development involves two aspects [De Reffye 2003]: the appearance of new metamers (on the same axis from an apical bud or on starting a new lateral axis from a lateral bud), and the geometrical expansion of the existing metamers (usually short in time). This second aspect is mainly related to environmental conditions, while the first one is essentially caused by genetics.

New metamers appear usually in a group (the growth unit) with various characters in terms of viability and branching potential; biologists show that this potential is related to the physiological age. In practice, 2 to 4 physiological age steps explain drastic metamorphosis in plant developments and describe the basic pattern in trees: the “architectural model” [De Reffye 1988, Barthelemy 2007]. Physiological age can be seen as a morphogenetical status starting from high development potential (age 1) to low ones (transformation to flower for instance). In classical approaches, tree generators define implicitly different characters from the

ramification order. To summarize, a plant can be generated from rules depending on a physiological age (or branching order), adding new sets of metamers (growth units) from end of existing axis or from metamers leaf insertions. Geometrical aspects (angles, length) can also be associated to this physiological age index.

2.1.2 GreenLab's Dual Scale Automaton and Substructure

For our biologically inspired model, Xing Zhao [Zhao, 2003] has shown that a dual scale automaton can be used to described and simulate any plant architecture. The deepest level describes the metamer successions (the micro states) in a typical growth unit at a given physiological age. The second level describes unit successions (the macro states) with possible physiological age changes, even on the same axis.

In the GreenLab model, all macrostates of a given physiological age born at the same date are similar (sharing the same attributes and the same fate) and define a subtree called a substructure by Yan et al. [Yan 2002]. Their evolution can be computed only once.

2.1.3 Encoding Dual-Scale Automaton and Sub Structures with L_systems

L-system is a grammar rule string rewriting system. Starting from a initial string (axiom), letters are replaced by production rules by words using the same alphabet, building a new string. We translate the concept of Microstate in Dual-Scale Automaton into a L-system, using the following alphabet:

Bi Bud at physiological age i ($i \geq 0$)

Mi Metamere at physiological age i ($i \geq 0$)

[] Used to start a new axis, as defined in classical L_systems

The general description of Microstate (sequence in a growth unit) can then be written as follow:

$Bi \rightarrow Mi ([B_j] \dots) B_k$ ($j, k \geq i \geq 0$). We note here “(...)” to say that there could be none or several lateral buds. Constraints on physiological age ($j, k \geq i$) translates the biological assumption that the physiological age of a new element is always higher than than of its father. The *Mi* letters stand for internodes of the tree, and “[”, (respectively “]”), stack (respectively unstack) a sub-tree from a lateral bud.

Macrostates (growth units) evolution is performed by duplicating microstates:

Macrostate i (n) {list of Microstates} where i stands is the macrostate physiological age, n the number of cycles the macrostate is replicated until it is transformed to Macrostate $i+1$.

Applying these rules, the resulting string may reach a huge size very quickly but that can be reduced thanks to sub-structuring consideration. Let's note *Rit*, a bud *Bi* with i as its physiological age and t as its birth time. Now, at cycle t , before rewriting the tree string, any bud appearing with physiological age i is replaced by *Rit*. *Rit* is associated to a string *Sij*. Rewriting does not affect *Rij*, but affects *Sij*. The principle applies recursively on all *Sij* strings. The number of *Rit* instances, called multiplicity, is recorded accordingly.

2.2 The Tree Graph: a Simple, Compact Tree Structure Encoding

The proposed encoded virtual tree structure is based on a classical graph description where nodes stand for the tree nodes : the metamers and links for the topological relations between them, similar to [Lintermann and Deussen 1999] work. Original point is that each node is also given time and age attributes.

2.2.1 Principles: Using the Metamer Description Level

A graph description build from metamere nodes can easily be designed from its sons; in our case we differentiate the son on the same axis from the lateral ones. We have implemented this graph from a simple list of nodes. Each node is given an id, its number of sons, the node id of the son on the axis (set to -1 if not existing) and the list of the lateral son(s) node(s) ids. We illustrate here an example of a single binary tree (Leeuwenberg tree model) of age 2, with units also defined by 2 microstates (Figure 1).



Figure 1. A simple binary tree

Using the L_system approach as defined earlier, the corresponding grammar for this model is:

Macrostate 1 (1) B0 -> M0 B0
B0 -> M0 [B0] [B0] B0
Macrostate 2 (1) B1 -> F (F stands for a flower)

At age 2 (ie after two iterations), the corresponding string is:

M0 M0 [M0 M0 [B0] [B0] B0] [M0 M0 [B0] [B0] B0] F

Assuming that each node is described on a line, the resulting topological encoding can be written as :

Node id	Number of sons	Successor	Axillaries	
0	1	1		
1	3	-1	2	4
2	1	3		
3	3	-1	-1	-1
4	1	5		
5	3	-1	-1	-1 (*)

(*) an alternate way could be to specify no sons at all

This encoding leads to a tree is easy to define for nearly all kind of ramified patterns, created from generators or simply measured on real world objects.

2.2.2 Efficient Topology Encoding: Avoiding Redundancies

We have shown that in most of procedural generated virtual plants, the simulators generate similar patterns at a given cycle(duplicate nodes). In our graph, we will just refer to the same.

On our exemple using the L_system approach, the corresponding grammar for this model is of course still the same but at age 2 (ie after two iterations), the corresponding string can now be written as:

Macrostate 1 (5) { M0 M0 [R01] [R01] F
S01 (2): M0 M0 [B0] [B0] B0 }

where *S01* stands for the substructure *R01*. In our typology we chose explicit time and aging; the substructure comes from a bud at physiological age 0 and did appear at year 1. The number (2) in the bracket is the multiplicity of this substructure.

The resulting topological encoding can now be written as (-1 as node id means no successor):

Node id	Number of sons	Node successor	Axillary Nodes	
0	1	1		
1	3	-1	2	2
2	1	3		
3	3	-1	-1	-1

2.2.3 Encoding Geometrical Metamer Properties

Each graph node can also be attached geometrical attributes. It is understood that in case of multiplicity, these attributes are the same for each instance, except origin, initial direction and mechanical behavior. Typical geometrical parameters are the length and the diameter. In our implementation we chose to encode volumes, since our plant model is a functional one and computes masses. More precisely we encode the pith volume and the rings volume of the corresponding metamer. Since each metamer carries leaves (see 2.1 section), we add the leaf volume. Note that the branching and phyllotaxy angles are not stored, they are implicit to physiological age.

2.2.4 Encoding Time and Aging Metamer Properties

In many plant simulators, aging attributes can be set from the generator. The time of apparition (or the cycle) can be stored as a birth date. Nodes typology such as physiological age is also an aging attribute to which additional parameters such as angles or graphical material properties are related.

3. RECONSTRUCTION AND VISUALIZATION

We can reconstruct the simulated plant from its encoded graph simply by defining branching and phyllotaxy angles and traversing the graph. Moreover, with time and aging attributes, the encoded graph allows to generate all past growth stages.

3.1 A Fast and Easy Reconstruction Way

3.1.1 The Reconstruction Pseudo Code

When geometrical representation is requested, the tree graph and its attributes are loaded in memory. The tree graph is a classical oriented graph, that can be traversed in several ways, in prefix order or not, using a stack or a recursive implementation. In our case, we traverse the graph by oldest node first, axis by axis. In other words starting by the trunk, order by order. We chose this way in order to generate easily Level of Detail models: merging metamers of the same direction together, or stopping rendering at a given branching order.

A pseudo code algorithm is given below (non recursive):

```

stack ← first node // first node is the tree origin, at bottom trunk
while (stack not empty) {
    current node ← stack
    if (node already seen)
        draw substructure generated from current node
    else {
        while (current node exists) {
            compute node position and draw (export) node geometry
            set node as seen
            for each lateral node
                compute lateral node orientation and stack the lateral node
            current node ← next node //on the same branch
        }
    }
    while (top of stack node seen)
        unstack
}
```

3.1.2 Building the Tree Geometry

A fast way to define the geometry is to assign a unit length to each node. For the diameter, a polynomial law function of metamer rank can be used [De Reffye 1988]. Lengths can also be parametrized from physiological age or correlated to the existence of lateral nodes (usually shorter). Usually length and diameters values are computed during tree generation and can be attached to each node attribute. In functional structural models, biomass volumes are defined for each node (for the pith and the rings). Using allometric rules, length can be deduced as well as diameters (See annex for this conversion).

Lateral nodes orientation and position is set according to the father node direction with two angle values: the insertion angle and the phyllotaxy angle. Both can be considered as related to the physiological age. These values are therefor not interesting to set as node attributes, since they are the same for each node of same physiological age (1 to 4 different values for each angle for the full tree).

Once node positions and orientation are known, exports to various format can be performed, generating components display lists [Jaeger 1992] or a skeleton base with its radii for the woody part.

Note that, in the case of substructures, this formalism takes its full advantage when orientations and positions are computed relative to the father node. Geometrical operations can then be encoded as affine transformations, easy to convert to graphical libraries instructions. On first call, substructures can be stored as nested graphical object definitions, to be instantiated when substructures need to be drawn again.

3.1.3 Leaves

As specified in the previous section, each node (metamer) carries leaves, and lateral buds grow at insertion of leaves. The orientation and the position of leaves on the axis is thus clearly defined: on the top of the metamer, with the orientation of the lateral axis. In practice, in order to avoid self collisions, the insertion angle is smoothly increased, and the leaf origin pushed to the branch surface (not the center). Similar to node length, the organ size can be defined from an empirical law, or computed from the simulator.

This approach can be extended to all organs: flowers and fruits. An alternate way is to specify a type attribute to each node (for instance in the *L_system* examples in section 2, production involves the letter F for flower apparition) and thus allow the simulator to allow specific location for organs.

However, aging is a key point for organs, since they are usually not persistent. As described below, self pruning can easily be performed when node birth date is given as an attribute.

3.2 Building a Wide Range of Tree Geometries from a Single Tree Graph

So far, given a tree graph computed at a given age and a limited set of parameters (length law, branching and phyllotaxy angles, drawing organs or not), a user can interactively design several geometries for the tree and generate display or exports. Taking advantage of aging and time information in node properties, we can produce new geometrical outputs, using the temporal dynamics of plant development.

3.2.1 Self Pruning

Self pruning is easy to implement. Let us first consider the specific case of leaf organs. Tree species growing in temperate areas have leaves that are alive during one single cycle (one year); leaf display is thus decided based on a single threshold on the birth date. More generally, self pruning can appear after a given pruning period following death. During this period, geometrical or material properties attached to the organ can be different in order to simulate metamorphosis (from flower to fruit) or senescence. For a given tree age, a death status can be computed as:

$$\text{death status} = \text{tree birth date} + \text{tree age} - \text{organ birth date} - \text{organ life span} + \text{pruning period}$$

If the organ is still alive, the status is positive, the geometry has to be generated. If the status is greater than the pruning period, the organ is functional, otherwise it is not -but still on the tree-, its color or shape attributes may be changed. If the status is negative, no geometry has to be generated. An example of such a pruning is shown in Figure 2, leaves are getting darker before getting pruned on older stage.

The method can be applied on nodes as well. In this case the major difficulty lies in the definition of the axis life span. This information can be retrieved exploring the substructure born from the node, tracking the nearest birth date of the substructure. This is easy to realize but can be a little bit costly. In our implementation we proceed in two steps. We retrieve first the life span for a given physiological age (this is supposed to be constant), since this period is supposed to decrease with the age value, the substructure exploration can be optimized. Application of self pruning on axis is illustrated on Figure 4 (axes drawn in light gray) and on Figures 5 and 6 where small dead branches on older trees are pruned.



Figure 2. Leaves life span and pruning

3.2.2 Growth Animation

The basic idea is to explore the graph with constraints over the birth date. The reconstruction algorithm remains the same, including the self pruning effect, just ignoring nodes born after the specified date, and setting the final tree age to the appropriate one. For the geometrical evolution, we suppose that the expansion duration is one growth cycle (such is the case for temperate trees). Under this condition, node length is stable as well as leaf volumes (ie areas). But axes diameters are not, they increase during tree growth because of ring accumulation. We use here a simple quadratic interpolation from metamer age to its final age. Figures 4, 5 and 6 show the interest of this approach. Two different tree graphs are used in figures 4 and 5, and both graphs used in Figure 6.

4. RESULTS AND PERFORMANCES

4.1 Architectural Models

As specified in section 2.1, plant topology and geometry is defined by its architectural model. Figure 3 show that the proposed L_System based on the dual scale approach is able to generate such architecture. An encoding graph does not manage the geometrical arrangement of the axis, such as the phyllotaxy. Using this approach, there is no difference between Rauh and Massart models. They differ in the spatial organization of the branches, orthotropic for Rauh and lying in the plane for Massart, managed in the reconstruction tool.



Figure 3. Simulation of three architectural models. From left to right Leeuwenberg, Rauh, and Massart models

4.2 Simulated Plant Examples and Application to Landscape Scenes

The approach is applied here on two “realistic” cases and used in a small forest scene. Geometrical export defines a component display list as specified in [Jaeger, 1992]. Four components are generated: woody parts (one per node standing for branch segments) and 3 kinds of leaf components, sharing the same geometry but with different graphical materials (diffuse, specular and rugosity contributions) according to their age. Each node in the graph satisfying the condition on age thus generates a set of wood components (whose cardinal is equal to the node multiplicity) and its corresponding leaves (if not pruned). Realistic view is generated from a classical polygonal OpenGL based renderer with shadow depth maps, fog and depth of field.

The first example reproduces the example given in [De Reffye, 2003]. The generator is a simple L_system production tool. In this example, an unique plant of age 8 is computed, and no geometrical attributes were simulated. All nodes show the same length. An empirical law defines the final diameter from the plant age and the node age. Phyllotaxy is 147.5 degrees and insertion angles varies from 60 to 75 degrees according to the branching node physiological age.



Figure 4. A realistic case. Left, flat sketch views plant reconstruction at age 2, 3 and 4. (grey levels stands for physiological ages). Right, rendered plant reconstruction with leaves at age 7 and phyllotaxy angles set to 147.5 degrees.

The second example illustrates a structural functional simulation output, using the continuous GreenLab simulator [Li, 2010]. The tree was computed with a daily simulation (no discrete growth cycles) for a duration of 245 months (20 years and a half). Time and aging attributes are thus expressed in floating point values. Node length and diameter are computed from volumes as defined in section 3.



Figure 5. A structural functional tree. From left to right, the five first views show the tree reconstruction and rendering at ages 3, 5, 7, 8, and 10. Right, realistic view at age 11, with an smaller allometry factor, leading to thinner axes.

We finally build a small scene using both encoded structures. The scene contains nearly 200 trees corresponding to 14 different trees. In fact only two tree graphs were used to build six reconstructions from ages 3 to 9 of the first plant (see Figure 4) and 8 reconstructions (Figure 5) of the second tree. The scene description is a simple ascii file listing the positions and rotations of the various plants, similar to a VRML with transform nodes and inline calls. The full description scene is therefore very compact. The 3D renderer is adapted from a simple polygonal rendering tool, calling the reconstruction tool on the first instance of each plant. Shadow fog and depth of field effects are post processed using depth maps techniques (Figure 6).

4.3 Performances

In our examples, graph generation has no measurable memory and CPU cost, since close to inner structures generation. On our examples, reconstruction user time is low, it reaches 2 seconds on the biggest tree (Figure 5, at age 50), mainly due to export to disk.



Figure 6. A small natural scene build from a tree position list and two tree graphs.

We summarize the performance gained on several examples shown in this paper in Table 1. It contains the graph file size and the number of nodes (metamers). The age of the reconstructed tree is then given together with the final simulated age. The number of components (internodes and leaves) of the reconstruction is given. Line Tree space is indicative, and could be understood similar to a geometrical polygonal size storing 50 to 80 bytes per component. The sequence size is the size of all Line trees generated from the tree graph from age 1 to final age.

Table 1. Tree graph size versus geometrical output size.

Tree model	Graph size	Nodes	Age vs Final age	Components	Line tree size	Sequence size
Fig. 2 right	2.0 Kb	225	6 on 10	26162	1.2 Mb	32 Mb
Fig. 4 right	2.2 Kb	252	7 on 10	5570	251 Kb	2.5 Mb
Fig. 5 middle	16.7 Kb	2006	7 on 19	3287	156 Kb	23.8 Mb
Fig. 5 right	16.7 Kb	2006	10 on 19	10049	474 Kb	23.8 Mb
Fig. 5 (+)	18.9 Kb	2258	19 on 19	59161	2.54 Mb	23.8 Mb
Fig. 5 (++)	96.7 Kb	8234	50 on 50	1798357	65.7 Mb	918 Mb

(+) Tree in fig. 5 was reconstructed at age 19 but not shown in the paper.

(++) Tree in Fig. 5 was simulated and reconstructed at age 50 but not shown in the paper.

Comparing graph sizes and components number show that substructure encoding leads to drastic compression. In Fig 6, the size of the scene description is close to 60 Kb, including the two graph size. The total number of the components is over 3 millions.

5. CONCLUSION

Virtual tree geometries have high bandwidth and rendering costs when natural scenes are to be displayed on client machines. Compared to image based techniques, rule based approaches seems to show new interests but lead to heavy specific developments on the client. However, we underline that rule based virtual plants approaches share common basic elements in their definition of virtual plants. Elementary patterns, rules of developments leading to the addition of new components, and, indeed, duplicated similar structures.

We proposed a structure encoding focused on the plant topology and these elementary patterns. We described an efficient tree graph structure to encode rule based virtual plants, suitable to various plant

generators: L_systems, dual scale automaton, patterns. Based on node links (related to biological metamer notion), the graph avoids similar structure encoding, referring simply to a tree axis typology and the birth date (cycle) of the first element of the sub tree. We showed how such a structure can be implemented with grammars and gave the generic reconstruction frame. Adding time and aging attributes allows to reconstruct geometry of the plant at any growth stage, in a simple, fast graph exploration. Organ reconstruction is easy to add including aging effects such as self pruning.

The approach is suitable for a wide range of applications from multimedia applications to scientific studies; and is really efficient for Web applications.

However, since the approach takes advantage of inner similarities, geometrical deformations related to absolute position or orientations are ignored. Such are the cases of mechanical behaviors (bendings, twists, ...). These deformations should be post processed, perhaps using the GPU capabilities.

Diameter evaluation while reconstructing the former ages is currently performed using an interpolation, giving graphical satisfaction, but biologically not founded. Rigorously, the diameter is function of the past accumulated leaf area, that can be evaluated from the graph but at a higher cost.

Finally, we are currently exploring two interesting tracks using this tree graph.

The first one is to use the graph exploration as a LoD tool: attributes such as physiological age, or number of nodes or volume of the underlying substructure gives interesting criteria in terms of feature importance. The second one is even more captivating: environmental pressure, on small cultivated plants mainly changes the production, and the development speed, but the topology remains quite stable. It means, that, for cultivated plants such as rice, wheat, sun flower... the same tree graph could be used with various environmental conditions. Qualitative variability in virtual crops could thus be efficiently visualized.

ACKNOWLEDGEMENT

This work is supported by the following projects: ARCUS ChongMing Agropolis Region Languedoc Roussillon, and french National program support ANR in project ANR 3Words

REFERENCES

- Barthelemy, D. and Caraglio, Y., 2007. Plant architecture: a dynamic, multilevel and comprehensive approach to plant form, structure and ontogeny. In: *Annals of Botany*, Vol.1, No 33, pp. :1-33.
- Brownbill, A. 1998. Reducing the storage required to render L-system based models, PhD thesis of University of Calgary. <http://dspace.ucalgary.ca/bitstream/1880/29367/1/20821Brownbill.pdf> Calgary, Alta., Canada, 134 p.
- Deng Q, Zhang X, Gay S, Lei X. 2007. Continuous lod model of coniferous foliage. *International Journal of Virtual Reality*, 6(4): 77–84.
- Deng, Q. and Zhang, X. and YANG, G. and Jaeger, M. 2010. Multiresolution foliage for forest rendering, *COMPUTER ANIMATION AND VIRTUAL WORLDS*, 20, 1, pp. 1-23
- De Reffye, P. and Jaeger, M. and Edelin, C. and Françon, J. and Puech, C. 1988. Plant models faithful to botanical structure and development. *Proceedings of Computer Graphics Siggraph 1998*, Atlanta, USA, 22, p. 151-158.
- De Reffye, P. and Goursat, M. and Quadrat J.P. and Hu, B., 2003. The dynamic equations of the tree morphogenesis GreenLab model. *Proceedings of Plant Modeling and Applications (PMA03)*. Beijing, Springer Tsinghua, 2003. pp. 108-117
- Deussen, O. and Lintermann, B. 1997. A modelling method and user interface for creating plants. *Proceedings of Graphics Interface 97*, Morgan Kaufmann Publishers, pp. 189-197
- Deussen, O. and Lintermann, B. 2004 *Digital Design of Nature: Computer Generated Plants and Organics*. SpringerVerlag.
- Fernández, I.G.. 2007. *Generation and Interactive Visualization of 3D vegetation*. Master Thesis, Master in Computing. http://www.gametools.org/archives/publications/mastertesi_isma.pdf Universitat de Girona, 2007.
- Jaeger, M. and De Reffye, P. 1992. Basic concepts of computer simulation of plant growth. *Journal of Biosciences*, 17, 3, p. 275-291.
- Li, Z and Le Chevalier, V. and Courbède, P.H., 2010, Towards a Continuous Approach of Functional-Structural Plant Growth, *Proceedings of Plant Modeling and Applications (PMA09)*. Beijing, IEEE, CPS, pp. 334-340

- Lintermann, B. and Deussen, O. 1999. Interactive structural and geometrical modeling of plants. *IEEE Computer Graphics and Applications*, 19, 1 pp. 56-65
- Max, N and Deussen, O and Keating, B. 1999. Hierarchical image-based rendering using texture mapping hardware. *Proceedings of the 1999 Eurographics Workshop on Rendering*, 1999, p. 57–62.
- Nevill-Manning, C.G. and Witten, I.H. and Mulsby, D.L. 1994. Compression by induction of hierarchical grammars. *Proceedings of IEEE Data Compression Conference DCC94*. Publisher: IEEE Comput. Soc. Press, Pages: 244-253
- Prusinkiewicz, P. and Lindenmayer, A. 1990. *The Algorithmic Beauty of Plants*. Springer Verlag, New York, NY, USA.
- Sun, R. and Jia, J. and Li, H. and Jaeger, M. 2009. Image-based lightweight tree modeling. *Proceedings of VRCAI '09*, New York, NY, USA: ACM, 2009. pp. 17-22.
- Sun, R. and Jia, J. and Jaeger, M., 2009. Intelligent tree modeling based on L-system. *Proceedings of CAID and CD'2009*. IEEE Computer Society, 2009. pp. 1096-1100.
- Zhao, X. and de Reffye, P. and Barthelemy, D. and Hu, B., 2003. Interactive simulation of plant architecture based on a dual-scale automaton model. *Proceedings of Plant Modeling and Applications (PMA03)*. Beijing, Springer Tsinghua, 2003. pp. 144-153.
- Yan, H. and Barcz, J.F. and De Reffye, P. and Hu, B. and Jaeger, M. and Le Roux, J. Fast Algorithms of Plant Computation Based on Substructure Instances. 2002 *Proceedings of WSCG'2002 – the 10-th International Conference in Central Europe on Computer Graphics, Visualization and Computer. Vision'2002* in cooperation with EUROGRAPHICS. Vol. 3, no. 10, 145-153

Annex

Volume to length/section conversion is performed thru following equations:

Definitions

V_p : volume of pith

V_r : volume of rings

H : internode height

R_i : internode radius

R_p : pith radius

Parameters for pith allometry are :

H_{ref} : reference length (set to 5 in our application)

V_{ref} : reference volume (set to $P * H_{ref}$ so nearly 15.71)

b : allometry parameter (set to 1 in our application)

Dimensions from volumes conversions:

$$H = H_{ref} \cdot (V_p / V_{ref})^{b/(1+b)}$$

$$R_p^2 = (V_{ref} / P \cdot H_{ref}) \cdot (V_p / V_{ref})^{1/(1+b)}$$

$$R_i^2 = R_p^2 + (V_r / P \cdot H_{ref}) = (V_{ref} / P \cdot H_{ref}) \cdot (V_p / V_{ref})^{1/(1+b)} \cdot (1 + V_{ref} / V_p)$$

